2002

Design techniques for high-performance current-steering digital-to-analog converters

Yonghua Cong
Iowa State University

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Design techniques for high-performance current-steering digital-to-analog converters

by

Yonghua Cong

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee
Randall L. Geiger, Major Professor
Degang Chen
Robert J. Weber
Chris Chong-Nuen Chu
Yuhong Yang

Iowa State University

Ames, Iowa

2002

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Yonghua Cong

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

ABSTRACT

Digital-to-Analog Converter (DAC) is a crucial building block limiting the accuracy and speed of many signal processing and telecommunication systems. To achieve high speed and high resolution, the current-steering architecture is almost exclusively used. Three important issues for current-steering DAC design are addressed in this dissertation. In a current-steering DAC design, it is essential that a designer determine the minimum required current source accuracy to overcome random current mismatch and achieve high linearity with guaranteed yield. Simple formulas are derived that clearly exhibit the relationship between the standard deviation of unit current sources, the bits of resolution, the INL/DNL, and the soft yield of DAC arrays. It is shown that these formulas are very effective for optimizing the DAC segmentation so as to achieve high performance and high yield with minimal area and power consumption. To overcome random mismatch effects without any trimming, the current source array of a high-accuracy DAC is usually rather large, causing the gradient errors in these arrays to become significant. How gradient errors affect the DAC linearity and how to compensate for them through switching sequence optimization is analyzed in the second part of this dissertation. To overcome technology barriers, relax the requirements on layout and reduce the sensitivities of DACs to process, temperature and aging, calibration is emerging as an attractive solution for the next-generation high-performance DACs, especially as process feature size keeps shrinking and supply voltage is reduced correspondingly. A new foreground calibration technique suitable for low-voltage environment is presented in the third part of this dissertation. It can effectively compensate for current source mismatches, and achieve high linearity with small die size and low power consumption. The dynamic performance of the DAC is also improved due to the dramatic

reduction of parasitic effects. To demonstrate this technique, a 14-bit prototype was designed and fabricated in a 0.13u digital CMOS process. It is the first 14-bit CMOS DAC ever reported that operates with a single 1.5V power supply, occupies an active area less than $0.1mm^2$, and requires only 16.7mW at 100MHz sampling rate, but still maintains state-of-art linearity and speed.

CHAPTER 1

OVERVIEW

In many signal processing and telecommunication applications, the digital-to-analog converter(DAC) is a crucial building block limiting the accuracy and speed of the overall system. When applications require high speed and high resolution, the current-steering DAC is almost exclusively used [1]-[5]. The basic idea of the current-steering DAC is very straightforward: Figure 1.1 shows a typical current-steering DAC where current sources are

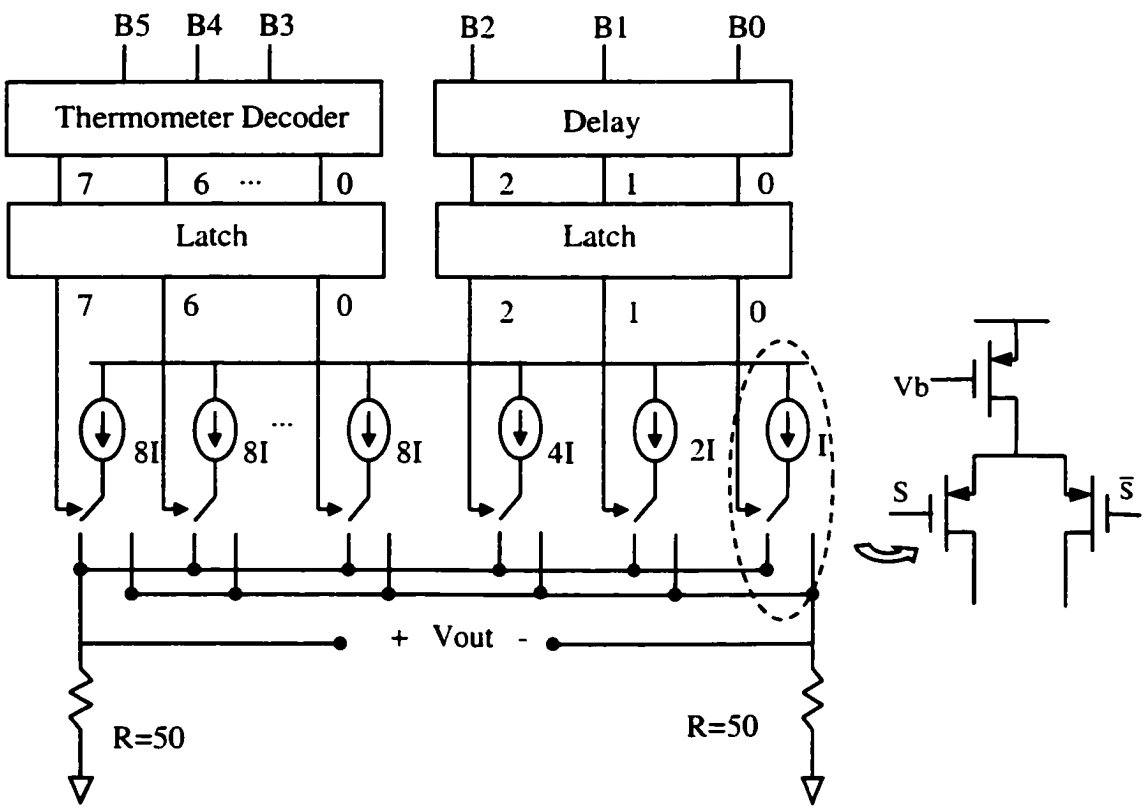


Figure 1.1 A 6-bit (3+3) segmented current-steering DAC architecture

implemented with PMOS transistors. Conceptually, assume for n-bit resolution, the array consists of N ($=2^n-1$) identical current sources, each providing a constant current I. The current I is termed 1 LSB (Least Significant Bit). Driven by the digital input, D ($0 \le D \le N$), D current sources are output to an external resistor, R (typically R=50 Ω), generating a voltage output, $V_o = I \cdot R \cdot D$. Thus, Vo is a linear function of D. If the remaining current sources are switched to another external resistor R simultaneously, differential outputs can be obtained with a differential voltage equal to $I \cdot R \cdot (2D-N)$ and common-mode voltage equal to $I \cdot R \cdot N/2$. Note that the differential output voltage is still a linear function of D while the gain is doubled compared to the single-ended case and so is the output swing, which is now from $-I \cdot R \cdot N$ to $I \cdot R \cdot N$. Other well-known advantages of differential structure include cancellation of even-order harmonics, high immunity to supply noise and other common-mode noise, and so forth. Therefore, differential structures are preferably used in most DAC design.

In the above, we assumed all of the current sources are identical and the currents they provide are equal and constant. We also assumed that when the digital input changes, the analog output changes instantly with a sharp and neat transition. However, these assumptions are not valid in the real world. Many nonidealities may impact the DAC performance. For example, due to process and temperature variations, mismatch always exits between the current sources. The currents may also vary with the output voltage due to the finite output impedance of the current sources. Besides, the nonideality during a transition, such as finite settling time, glitch, timing skew etc. may introduce undesired harmonics in the output spectrum.

The performance of current-steering DACs is generally characterized by their static and dynamic properties[6]-[8]. The static properties represent the DAC performance at dc or low frequencies. They are determined by the settled analog output and generally measured by the offset, gain error, and nonlinearity. The dynamic properties of a DAC are affected not

only by the settled values but also by the transition between two successive output states. In the time domain, the transient behavior is measured by settling time, glitch energy, etc, while in the frequency domain, the DAC performance is measured by spurious-free dynamic range (SFDR), signal-to-noise ratio (SNR), total harmonic distortion (THD), intermodulation distortion (IMD), etc. Linearity is often the most important concern for current-steering DACs, and is impacted by both static nonlinear errors and the nonlinear transient behavior. Which one dominates is dependent on the input signal frequency, the conversion rates and the DAC architecture.

In the following, an brief overview of the major error sources causing static and dynamic nonlinearities will be given first, and then the conventional DAC architectures and their influence on the DAC linearity will be discussed next.

1.1 STATIC LINEARITY AND MAJOR ERROR SOURCES

Static linearity is typically measured by integral and differential nonlinearity (INL and DNL) [7]. After the offset and gain are corrected, the end-point INL at digital code D, is defined as the deviation of the real analog output, I(D) in the units of LSB, from the ideal output, D, while the DNL is the deviation of each step size from 1 LSB. The INL and DNL of an n-bit DAC array can be expressed as

$$INL(D)=\frac{I(D)-I(0)}{[I(N)-I(0)]/N}-D \tag{1.1a}$$

$$DNL(D)=\frac{I(D)-I(D-1)}{[I(N)-I(0)]/N}-1 \tag{1.1b}$$

where $N=2^n-1$ and n is the bits of resolution. For differential output, simply substitute I(D) with $Id(D) = I(D) - \overline{I(D)} = 2I(D) - I(N)$, hence Id(0)=-I(N) and Id(N)=I(N).

The major static error sources include mismatch between current sources and their finite output impedance, and will be discussed in the following two subsections respectively.

1.1.1 Current Source Mismatch

Due to random varieties during each processing step and environment variations, mismatch exists between any identically designed devices. The current mismatch in a typical unary current source array is shown in Figure 1.2 where the current of each individual current source was measured [5]. It can be seen that the mismatch errors distributed in the array can be distinguished into random errors and gradient errors. In this example, the gradient is approximately linear.

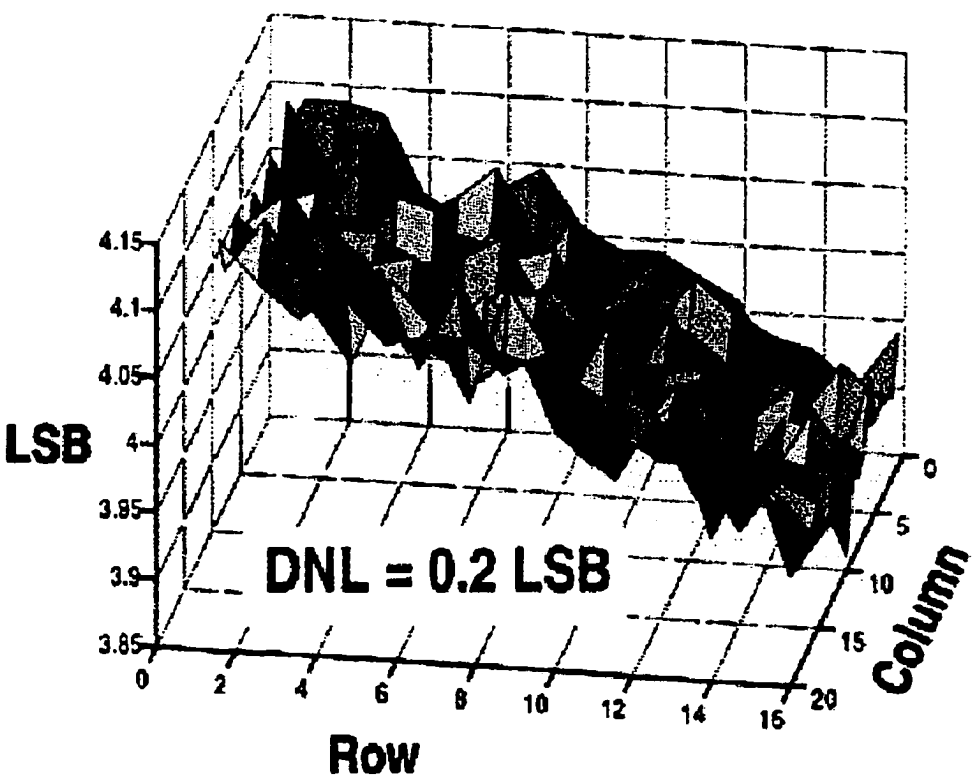


Figure 1.2 Mismatch between current sources in a typical unary array

In (1.1a&b), for single-ended output, I(0)=0 and I(N) is equal to the total current in the array, so [I(N)-I(0)]/N is actually the average current (denoted as $\bar{I}$) of the N unit current sources. Therefore, the INL and DNL at digital code D can be expressed as

$$INL(D) = \frac{I(D)}{\bar{I}} - D \tag{1.2a}$$

$$DNL(D) = \frac{I(D) - I(D-1)}{\bar{I}} - 1 \tag{1.2b}$$

It can be shown that (1.2a) and (1.2b) are also valid for differential outputs except that I(D) is substituted with $Id(D) = I(D) - \overline{I(D)}$.

A. Random Mismatch

Random mismatches are determined by the inherent matching properties of the technology used. Generally, random variations of devices are assumed to be uncorrelated and follow the Gaussion distribution. If the designed value of unit current sources is I, when only random mismatch is present, the actual current provided by the jth ($1 \leq j \leq N$) unit current source can be expressed as

$$I_j = I \cdot (1 + \varepsilon_j) \tag{1.3}$$

where $\varepsilon_j \sim N(0,\sigma^2)$ is the relative deviation of I_j from I.

Based on what is often termed the Pelgrom model [9], for a MOSFET in saturation region and with a certain overdrive voltage V_{GS}-Vt, the relative variance of the drain current can be expressed

$$\sigma^2\left(\frac{\Delta I}{I}\right) = \frac{A_\beta^2}{W \cdot L} + \frac{4A_{Vt}^2}{W \cdot L \cdot (V_{GS} - Vt)^2} \tag{1.4}$$

where A_β and A_{V_t} are the technology constants, and W and L are the width and length of the transistor gate respectively [10],[11]. An important conclusion drawn from this expression is that the current variation is inversely proportional to the gate area of the transistor. For a given process technology and for architectures that do not incorporate trimming or tuning, increasing the active area of each unit current source in the DAC array is the most effective method for reducing random errors.

If the required matching accuracy or the relative standard deviation σ is determined for each current source, it is easy to determine the gate area of the current sources from (1.4) for a given performance requirement. However, the relationship between the required matching accuracy of the current sources and the DAC linearity specifications such as INL, DNL is not straightforward. This topic will be discussed in detail in Chapter 2.

B. Gradient Errors

Many factors may cause gradients over a current source array. People observed that the spread of doping and oxide thickness over the wafer or the voltage drop along the power line can cause approximately linear gradient errors [12]-[15]. Temperature gradients and die stress may introduce approximately quadratic gradients [16]. The overall gradient error distribution is given by superimposing these error components. As it will be shown in Chapter 2, when only random errors are considered, for each extra bit of resolution, to maintain the same INL and yield, the active area of the unary array has to be increased by a factor of four. For high-accuracy DACs, for example a 12-bit intrinsic accuracy DAC, this may result in an array in the range of $1mm^2$ [13]. The gradient errors in these large arrays can become very significant and introduce large systematic errors. How gradient errors affect the DAC linearity and the methods to compensate for them will be discussed in Chapter 3.

1.1.2 Output Impedance

Even if the current sources are perfectly matched, the output current of each current source may still vary with the output voltage due to their finite output impedance, denoted as r_o [6],[8],[17]-[19]. For digital code D, D current sources are driven to a resistor of value R. The output impedance of the D current sources is equal to r_o/D. It is input dependant and so is the output current that can be expressed as

$$I(D) = I \cdot D - \frac{I(D) \cdot R}{r_o/D} \Rightarrow I(D) = \frac{I \cdot D}{1 + \rho D} \tag{1.5a}$$

where $\rho = R/r_o$. Similarly, the complementary output can be expressed as

$$\bar{I}(D) = \frac{I \cdot \overline{D}}{1 + \rho \overline{D}} \tag{1.5b}$$

where $\overline{D} = N - D$.

For single-ended output, replacing I(D) in (1.1a) and (1.1b) with (1.5a) and assuming the output impedance r_o is so high that $\rho D << 1$, we have

$$INL(D) = \frac{\rho}{1 + \rho D} D \cdot (N - D) \approx \rho D \cdot (N - D) \tag{1.6a}$$

$$DNL(D) = \frac{1 + \rho N}{(1 + \rho D)[1 + \rho (D - 1)]} - 1 \approx \rho N \tag{1.6b}$$

It is shown that the DNL due to finite output impedance is negligibly small while the maximum INL is approximately equal to $\rho N^2/4$ occurring at the mid-scale transition. Therefore, to keep the INL of a 14-bit DAC less than 0.5 LSB, r_o has to be at least 2^{26} higher than R, that is $r_o > 3.35G\Omega$, if $R = 50\Omega$.

This requirement can be relaxed by using differential outputs. In this case, the current output for digital code D is $Id(D) = I(D) - \bar{I}(D)$, where I(D) and $\bar{I}(D)$ are given in (1.5a) and (1.5b) respectively. Replacing I(D) with Id(D) in (1.1), for high r_o, we have

$$INL_d(D) = (\frac{N}{2} - D)\left[1 - \frac{1}{1 + \frac{\rho^2}{1+\rho N} D(N-D)}\right] \approx \rho^2 D(\frac{N}{2} - D)(N-D) \qquad (1.7a)$$

$$DNL_d(D) = \frac{1+\rho N}{2}\left[\frac{D}{1+\rho D} - \frac{N-D}{1+\rho(N-D)} - \frac{D-1}{1+\rho(D-1)} - \frac{N-D+1}{1+\rho(N-D+1)}\right] - 1 \approx (\rho N)^2 \qquad (1.7b)$$

The maximum INL is about equal to $\sqrt{3}\rho^2 N^3/36$ when D is close to $(1/2 \pm \sqrt{3}/6)N$. For the same 14-bit DAC as we mentioned previously, if differential output is used, to keep INL<0.5LSB, the minimum required output impedance of each unit current source is $1.24\times2^{19}\times R = 32.5M\Omega$, which is about 100 times lower than that is required for single-ended output! Apparently, the DNL in this case is also much smaller than that of the single-ended output.

Even if the requirement for output impedance can be much relaxed by using differential output, for high accuracy DAC, it is still hard to be achieved with a single transistor. Increasing the length of the transistors may help to enhance their output impedance, but to maintain the same full-scale current, hence the DAC output swing, the width of the transistors has to be increased in the same ratio. As a result, the area of the DAC array and the parasitic effects increase dramatically, limiting the conversion rate. On the other hand, the gradient effect becomes significant as well. As it will be described in Chapter 3, to compensate for the gradient errors, complex layout has to be employed, which further increases the delay due to long interconnections

Adding cascode stages is an effective method to enhance the output impedance of current sources [18]. However, for a certain supply voltage, adding a cascode stage reduces the effective gate-source voltage of the current source, hence degrades its matching property and lowers the immunity against noise and voltage fluctuation along the power line [17]. Especially as the feature size of CMOS technology continues to shrink in the favor of digital circuits and correspondingly the supply voltage is reduced to ensure the transistor reliability and reduce power consumption, it becomes more impractical to use multiple cascode stages. Therefore, to gain enough output impedance becomes one of the challenges for integrating high-performance DACs in fully digital low-voltage CMOS technologies

1.2 DYNAMIC LINEARITY AND MAJOR ERROR SOURCES

For video and wireless telecommunication applications, not only the static linearity but also the switching transient behaviors are important. In the frequency domain, they both contribute to harmonics that are the major factors limiting the spurious-free dynamic range (SFDR) of current-steering DACs [5],[18],[20],[21],[32],[38]. As high resolution reduces the quantization noise floor, the harmonics or spurs due to static and dynamic nonlinearities of a DAC do not scale down. As a matter of fact, they may become worse since more switches are employed.

The impacts of static error sources such as current mismatch and finite output impedance on the SFDR of a current-steering DAC have been well studied in [6] and [22], therefore will not be discussed in this thesis. As for the switching transition, nonlinearity may be caused by many reasons, such as insufficient settling, timing skew between the switch control signals, switch mismatch, clock feedthrough, voltage fluctuation at the internal nodes of current cells, nonlinear parasitic effect and so on [12],[18],[20]. Many strategies have been reported in the literature to solve these problems. For example, latches are placed right in

front of the switches to reduce the timing skew and improve the synchronization [12]-[14],[17],[19],[23]-[37],[49],[50]; switch drivers are used to avoid the two switches being turned off simultaneously and hence reduce the voltage fluctuation at the common node of the switches [12],[13],[17],[19],[23]-[37],[49],[50]; Cascode transistors can be added between the switches and the output to reduce the clock feedthrough through the gate-drain capacitance of the switches[13],[15]; A cascode transistor can also be added between the current source and the switches to reduce glitch due to the drain voltage variation of the current source [18].

The above error sources leading to dynamic nonlinearity are strongly dependent on the layout and the process used, and therefore the transient behaviors of a DAC during the switching are hard to model and predict. However, from another point of view, it can be found that like many other analog building blocks, the nonideal transition period of a current-steering DAC are mainly due to the existence of parasitic capacitance. As it was stated in [13], there are two nodes in a basic current cell shown in Figure 1.3. The poles associated

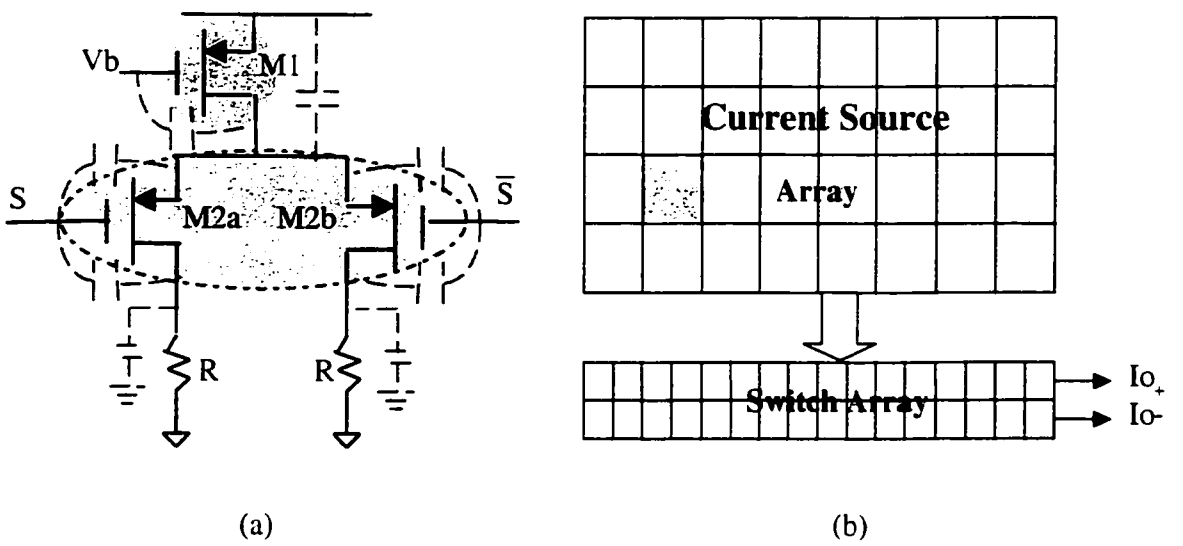


Figure 1.3 (a) A basic current cell (b) Layout of the current array

with them are

At the output nodes: $$p1 \approx \frac{1}{R_L(C_L + C_{dtot2})}$$

At the common node of the switches: $$p2 \approx \frac{g_{m2} + g_{mb2}}{C_{dtot1}}$$

where R_L is the doubly terminated 50Ω resistive load, C_L is the external load capacitance, C_{dtot1} and C_{dtot2} are the total parasitic capacitance at the two nodes. Insufficient settling, glitches due to the voltage fluctuation at the internal nodes of current cells, the decreasing of the output impedance at high frequency, and clock feedthrough, etc. are all related to the parasitic capacitance. Most of the time, these parasitic effects are input code dependant[6][18][22]. It is worth mentioning that in a high-resolution DAC, the parasitic capacitance seen at the drain of the current source(C_{dtot1}) is usually very large due to the large dimensional current source and the long routing between it and the switches. As we mentioned in Chapter 1.1.1, without trimming and calibration, large-dimensional current sources have to be used to overcome the random mismatch. Meanwhile, as shown in Figure 1.3, to make the current source array compact, and hence minimize the gradient effects, and to reduce the noise coupling from the digital circuits, the switches and decoding logic are placed outside the current source array, resulting in a long interconnection between them, especially when a complex switching scheme has to be used.

Therefore, as in many other analog circuits, the most effective method to improve the dynamic performance of a current-steering DAC is to reduce the parasitic effects and fast the settling. In this way, even if the switching transition is highly nonlinear, it is short compared to the overall conversion period, leaving the static nonlinearity dominant. A good way to achieve this goal is to use proper self-calibration. The details on this topic will be discussed in Chapter 4.

1.3 DECODING SCHEMES

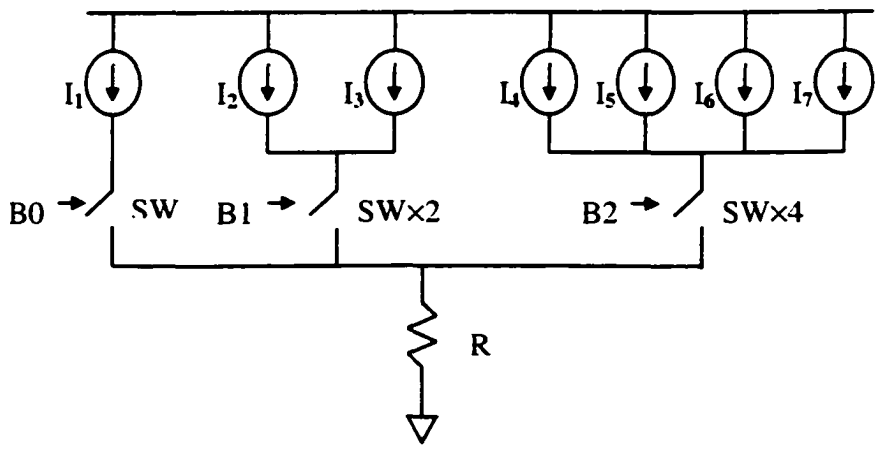
In the above discussion, we mentioned that for digital input D, D unit current sources are steered to the positive output, but we did not mention which D of the N current sources are selected. Actually, this selection is controlled by the decoding scheme of the DAC. Three decoding schemes are generally used in the literature, resulting in three DAC architectures: binary-weighted, thermometer-decoded and segmented DACs. It will be shown that decoding schemes affect both static and dynamic performance significantly. The INL and DNL of the three DAC architectures due to random mismatches and gradient errors are derived in Chapter 2 and Chapter 3 respectively.

The advantage of a binary-weighted DAC is its simplicity and low power consumption, as no decoding logic is required [5],[11],[31],[38],[44],[50]. However, this structure has drawbacks on both static and dynamic performance. These drawbacks are all associated with major carries of the DAC. The severity of the problem is proportional to the weight of the bit. The worst case happens at the mid-code transition when the MSB current source of the binary-weighted array needs to match the sum of all the other current sources. Therefore, this architecture is not guaranteed monotonic and may result in large DNL. As an example, Figure 1.4(a) shows a 3-bit binary-weighted DAC with 7 identically designed unit current sources. The current of each unit current source can be expressed by (1.3). To simplify the discussion, single ended configuration is used. At the mid-code transition (011→100), the output currents for 011 and 100 are equal to

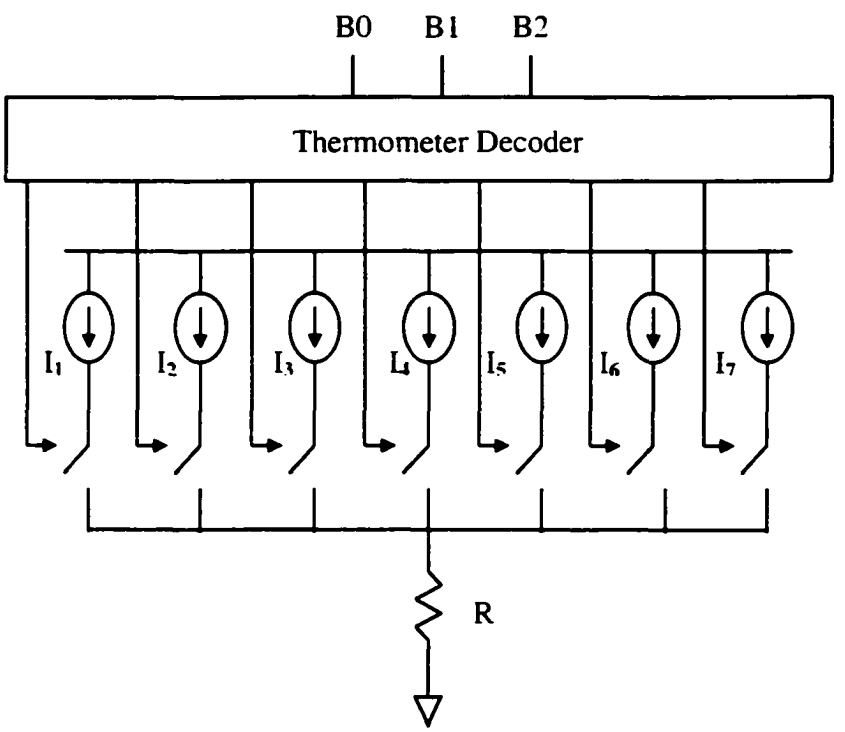
$$I(011) = I_1 + I_2 + I_3 = 3 \cdot I + I \cdot (\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$$

$$I(100) = I_4 + I_5 + I_6 + I_7 = 4 \cdot I + I \cdot (\varepsilon_4 + \varepsilon_5 + \varepsilon_6 + \varepsilon_7)$$

Based on (1.2a) and (1.2b), the INL and DNL at this transition are given by



(a)



(b)

Figure 1.4 (a) A simplified 3-bit binary-weighted DAC
(b) A simplified 3-bit thermometer-decoded DAC

$$INL(011)=\frac{I(011)}{\bar{I}}-3\approx\frac{4}{7}(\varepsilon_1+\varepsilon_2+\varepsilon_3)-\frac{3}{7}(\varepsilon_4+\varepsilon_5+\varepsilon_6+\varepsilon_7)$$

$$INL(100)=\frac{I(100)}{\bar{I}}-4\approx-\frac{4}{7}(\varepsilon_1+\varepsilon_2+\varepsilon_3)+\frac{3}{7}(\varepsilon_4+\varepsilon_5+\varepsilon_6+\varepsilon_7)$$

$$DNL(100-011)=\frac{I(100)-I(011)}{\bar{I}}-1=-\frac{8}{7}(\varepsilon_1+\varepsilon_2+\varepsilon_3)+\frac{6}{7}(\varepsilon_4+\varepsilon_5+\varepsilon_6+\varepsilon_7)$$

If the mismatches between the unit current sources are random and uncorrelated, and each current source has the same relative standard deviation, σ, then the standard deviation of INL and DNL at this transition are approximately equal to

$$\sigma_{INL}(011)=\sigma_{INL}(100)=\sigma\cdot\sqrt{(\frac{4}{7})^2\cdot 3+(\frac{3}{7})^2\cdot 4}\approx 1.3\sigma$$

$$\sigma_{DNL}(100-011)=2\sigma_{INL}\approx 2.6\sigma$$

On contrast, Figure 1.4 (b) shows a 3-bit thermometer decoded DAC. When the digital input increases by 1, one more current source is switched on. Thus, monotonicity is guaranteed. Using the same unit current sources as the binary-weighted DAC, the output currents of the thermometer decoded DAC at the mid-code transition (011→100) are equal to

$$I(011)=I_1+I_2+I_3=3\cdot I+I\cdot(\varepsilon_1+\varepsilon_2+\varepsilon_3)$$

$$I(100)=I_1+I_2+I_3+I_4=4\cdot I+I\cdot(\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_4)$$

The INL and DNL at this transition are given by

$$INL(011)=\frac{I(011)}{\bar{I}}-3\approx\frac{4}{7}(\varepsilon_1+\varepsilon_2+\varepsilon_3)-\frac{3}{7}(\varepsilon_4+\varepsilon_5+\varepsilon_6+\varepsilon_7)$$

$$INL(100)=\frac{I(100)}{\bar{I}}-4\approx\frac{3}{7}(\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_4)-\frac{4}{7}(\varepsilon_5+\varepsilon_6+\varepsilon_7)$$

$$DNL(100-011)=\frac{I(100)-I(011)}{\bar{I}}-1\approx\frac{6}{7}\varepsilon_4-\frac{1}{7}\cdot(\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_5+\varepsilon_6+\varepsilon_7)$$

The standard deviation of the INL and DNL in this case are approximately equal to

$$\sigma_{INL}(011)\approx\sigma_{INL}(100)=\sigma\cdot\sqrt{(\frac{4}{7})^2\cdot 3+(\frac{3}{7})^2\cdot 4}\approx 1.3\sigma$$

$$\sigma_{DNL}(100-011)=\sigma\sqrt{(\frac{6}{7})^2+\frac{6}{7^2}}\approx\sigma$$

Comparing the results for the two DACs, it can be seen that the INL is the same for both cases, however, the DNL of the thermometer decoded DAC is much smaller than that of the binary-weighted DAC. A generalized discussion of this issue will be given in Chapter 2.

Meanwhile, the two decoding schemes also result in different transition behaviors and hence different spectrum characteristics. In a binary-weighted DAC, during the mid-scale transition, all the current sources have to be switched on or off simultaneously, resulting in a large glitch and a large second-order harmonic [32]. Likewise, glitches at other carries are smaller according to the weight of the bits and result in higher-order harmonics. On the contrary, in a thermometer-decoded DAC, the current switched during each transition is minimized and proportional to the step size of the two successive digital inputs. As it was shown in [5], if there is no switch mismatch and the glitch is strictly proportional to the input step, it will not cause any nonlinearity except a little filtering effect.

The major drawbacks of thermometer-decode DAC are all due to the additionally needed binary-to-thermometer decoder, which costs extra area and power. For very high speed DAC, the design of this decoder is nontrivial and more stages of latches are often inserted for synchronization [19]. To take advantage of the high linearity of thermometer decoded DAC without costing too much area and power, segmented architecture as shown in Figure 1.1 is almost exclusively used in today's high-performance current-steering DAC

design. The performance of these DACs are mainly restricted by the thermometer-decoded array driven by the MSB's while the requirements for the binary-weighted LSB array are much more relaxed.

1.4 THESIS ORGANIZATION

Three important issues for high-performance current-steering DAC design are addressed in the next three chapters. As we described above, the performance of a current-steering DAC is primarily determined by the matching property of the current source array. In a current-steering DAC design, it is essential that a designer determine the minimum required standard deviation or the area of the current sources to overcome the random current mismatch and achieve given linearity specifications (INL and DNL) with guaranteed yield. This issue is discussed in Chapter 2, where statistical analysis are applied to both binary-weighted and thermometer-decoded DACs and simple formulas are derived to exhibit the relationship between the standard deviation of unit current sources, the bits of resolution, the INL/DNL and the yield of the DAC arrays. It is shown that these simple formulas are very effective for optimizing the segmentation and architecture of DACs so that high performance and high yield can be achieved with minimal area and power consumption.

To overcome random mismatches, the area of the current source array of a high-accuracy DAC turns out to be rather large. The gradient errors in these arrays become very significant and introduce large systematic errors. Chapter 3 analyzes how gradient errors affect the DAC linearity and how to compensate for them through switching sequence optimization.

Many reported high-performance DACs were achieved by pushing the designs to the technology limits. To overcome technology barriers, relax the requirements on the layout and reduce the sensitivities of DACs to process, temperature and aging, calibration is emerging as

a good solution for the next-generation high-performance DACs. Chapter 4 investigates calibration schemes suitable for high-accuracy, high-speed DACs operating in low-voltage environment. A new foreground calibration technique is presented which can effectively compensate for current source mismatches, and achieve high linearity with small die size and low power consumption. The dynamic performance of the DAC is also improved due to the dramatic reduction of parasitic effects. To demonstrate this technique, a 14-bit prototype was designed and fabricated in a 0.13u digital CMOS process. It is the first 14-bit CMOS DAC ever reported that operates with a single 1.5V power supply, occupies an active area less than $0.1mm^2$, and requires only 16.7mW at 100MHz sampling rate, but still maintains state-of-art linearity and speed.

CHAPTER 2

FORMULATION OF INL AND DNL YIELD ESTIMATION[1]

2.1 INTRODUCTION

As we mentioned in Chapter 1, random current source mismatch is a major error source leading to DAC nonlinearity. It is determined by the inherent properties of the technology used. In order to achieve a given linearity specification (INL and DNL) at a given yield level, it is essential that the designer decide the minimum required matching accuracy of the unit current sources. It is such a key parameter that determines almost all the features of the DAC. According to (1.4), the matching requirement determines the size of the transistor to be used, hence the size of the current source array, the gradient effects, the complexity of layout, the parasitic effects, the speed as well as the die size and power consumption.

Monte Carlo simulations are widely used in the literature [5],[11]-[13]. Assuming each unit current source has the same relative standard deviation σ, after one run of Monte Carlo simulation, the INL and DNL of an n-bit DAC can be calculated based on the definitions given by (1.1a&b). After thousands of runs, the yield of INL and DNL for this given σ can be obtained. Repeat the above simulations for different σ's, a plot of yield versus σ can be drawn for given INL and DNL specifications [11],[13]. These simulations are very time-consuming for high-resolution DAC's and the results are only useful for the given bits of resolution, the given DAC architecture and the given specifications. It may be tolerable compared to the whole design procedure when the DAC architecture is simple. However, it provides designers with little insight to understand how the matching accuracy affects the

linearity and yield when the bits of resolution and/or the segmentation change. These issues are very important for choosing proper DAC architectures and making tradeoff between different specifications, especially when calibration is used. To gain more insight, simple parametric expressions are preferred, which make it possible to optimize the DAC structure and achieve high performance with less cost and power consumption.

The limited mathematical formulations that appeared in the literature either oversimplified the statistical assumptions [40]-[42] or all based on nonstandard linearity definitions [43]. The former has resulted in a very rough lower and upper bounds for the relative standard deviation of unit current sources (σ), while the latter significantly overestimated σ leading to some questionable conclusions. An accurate estimation of σ is very critical, because as shown in (1.1), if σ is overestimated by a factor of 2, the gate area of the transistor or the current source array will be 4 times larger than it is required. On the other hand, if σ is underestimated by a factor of 2, as it will be shown, the yield of the DAC will decrease dramatically.

To investigate this problem, first, a statistical analysis of INL and DNL in thermometer-decoded and binary-weighted current-steering DAC's are presented. Following that, simple formulas are obtained that accurately describe the relationship between nonlinearity, bits of resolution, minimum required matching accuracy, and yield.

2.2 STATISTICAL CHARACTERISTICS

As mentioned in Section 1.1.1 of Chapter 1, in an n-bit DAC array consisting of $N=2^n-1$ identically designed unit current sources, if the designed value of the unit current is I, when only random mismatches present i.e. gradient effects and finite output impedance

effects are neglected, the actual current provided by the jth ($1 \le j \le N$) unit current source can be expressed as

$$I_j = I \cdot (1 + \varepsilon_j) \qquad (1 \le j \le N) \tag{2.1}$$

where ε_j is the relative deviation of I_j from I. It is approximately Gaussian distributed with zero mean and a standard deviation denoted as σ, i.e. $\varepsilon_j \sim N(0,\sigma^2)$. If the offset current I(0) is zero, the INL and DNL at digital code D can be expressed as

$$INL(D) = \frac{I(D)}{\bar{I}} - D \qquad (0 \le D \le N) \tag{2.2a}$$

$$DNL(D) = \frac{I(D) - I(D-1)}{\bar{I}} - 1 \qquad (1 \le D \le N) \tag{2.2b}$$

where $\bar{I}$ is the average current of the unit current sources as given by

$$\bar{I} = \frac{\sum_{j=1}^{N} I_j}{N} = I \cdot (1 + \frac{\sum_{j=1}^{N} \varepsilon_j}{N}) \tag{2.3}$$

A Thermometer-decoded DACs

If the DAC is thermometer decoded, then the output current at code D is equal to

$$I(D) = \begin{cases} 0 & (D = 0) \\ I \cdot (D + \sum_{j=1}^{D} \varepsilon_j) & (1 \le D \le N) \end{cases} \tag{2.4}$$

Through first-order approximation, (2.2a) and (2.2b) can be rewritten as

$$INL(D) \approx \begin{cases} 0 & (D = 0, N) \\ \frac{N-D}{N} \sum_{j=1}^{D} \varepsilon_j - \frac{D}{N} \sum_{j=D+1}^{N} \varepsilon_j & (1 \le D \le N-1) \end{cases} \tag{2.5a}$$

$$DNL(D) \approx \frac{N-1}{N}\varepsilon_D - \frac{1}{N}\sum_{j=1,\, j\neq D}^{N}\varepsilon_j \qquad (1 \le D \le N) \tag{2.5b}$$

Since INL(D) and DNL(D) are the linear combinations of N independent normal random variables ε_1 , ε_2 ,..., ε_N, they also follow normal distributions, and

$$INL(D) \sim N(0, \frac{(N-D)\cdot D}{N}\sigma^2) \qquad (0 \le D \le N) \tag{2.6a}$$

$$DNL(D) \sim N(0, \frac{N-1}{N}\sigma^2) \qquad (1 \le D \le N) \tag{2.6b}$$

The maximum standard deviation of INL is

$$\max(\sigma_{INL})\big|_{thermometer} = \frac{\sigma}{2}\sqrt{N-\frac{1}{N}} \tag{2.7}$$

occuring at the mid-code transition when D=(N-1)/2 or (N+1)/2. The DNL of a thermometer-decoded DAC is rather small and has approximately the same standard deviation as a single unit current source. In real designs where segmented architectures are exclusively used, the DNL of a DAC is determined by the binary-weighted DAC array, while its INL are mainly dependent on the thermometer-decoded array driven by the MSBs. Being lack of significance in practice, the DNL of thermometer-decoded DAC will not be further discussed.

The INLs at digital code 1,2,...,N-1 form a N-1-dimensional random variable that has a multi-variant normal distribution. This vector can be expressed as

$$\vec{\mathbf{INL}} = [INL(1), INL(2), \cdots, INL(N-1)] \approx \vec{\boldsymbol{\varepsilon}} \cdot \vec{\mathbf{B}} \tag{2.8}$$

where , $\vec{\boldsymbol{\varepsilon}} = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_N]$ and $\varepsilon_j \sim N(0,\sigma^2)$. According to (2.5a)

$$\vec{\mathbf{B}}=\frac{1}{N}\begin{bmatrix} N-1 & N-2 & \cdots & 2 & 1 \\ -1 & N-2 & \cdots & 2 & 1 \\ \vdots & -2 & & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & 2 & \vdots \\ -1 & -2 & \cdots & -(N-2) & 1 \\ -1 & -2 & \cdots & -(N-2) & -(N-1) \end{bmatrix}_{N\times N-1} \tag{2.9}$$

A simple normalization of $\overrightarrow{\mathbf{INL}}$ by the factor σ results in the expression

$$\overrightarrow{\mathbf{INLnor}}=\frac{1}{\sigma}\vec{\boldsymbol{\varepsilon}}\cdot\vec{\mathbf{B}} \tag{2.10}$$

since $$\frac{1}{\sigma}\vec{\boldsymbol{\varepsilon}} \sim N\left(\vec{\mathbf{0}}_{1\times N},\vec{\mathbf{I}}_{N\times N}\right)$$

where $\vec{\mathbf{0}}$ and $\vec{\mathbf{I}}$ represent the zero vector and the identity matrix respectively. It follows that

$$\overrightarrow{\mathbf{INLnor}} \sim N\left(\vec{\mathbf{0}}_{1\times N-1},\vec{\mathbf{B}}^{T}\cdot\vec{\mathbf{B}}\right) \tag{2.11}$$

where $\vec{\mathbf{B}}^{T}$ is the transposition of $\vec{\mathbf{B}}$. The 1×N-1 zero vector $\vec{\mathbf{0}}$ is the mean and the N-1×N-1 matrix $\vec{\mathbf{B}}^{T}\cdot\vec{\mathbf{B}}$ is the covariance of $\overrightarrow{\mathbf{INLnor}}$. With the mean and covariance matrix given above, it is easy to obtain the joint probability density function of $\overrightarrow{\mathbf{INLnor}}$, that is

$$f_{INL}(x_1,x_2,\cdots,x_{N-1})=\frac{1}{(2\pi)^{(N-1)/2}|\mathbf{C}|^{1/2}}\cdot\exp(-\frac{1}{2}\mathbf{X}\mathbf{C}^{-1}\mathbf{X}^{T}) \tag{2.12}$$

where $$\vec{\mathbf{C}}=\vec{\mathbf{B}}^{T}\cdot\vec{\mathbf{B}} \tag{2.13}$$

$$\vec{\mathbf{X}}=[x_1,x_2,\cdots,x_{N-1}] \tag{2.14}$$

$$x_D = \frac{INL(D)}{\sigma} \qquad \text{for } D=1,2,\ldots,N-1 \tag{2.15}$$

B. Binary-weighted DACs

A similar derivation can also applied to binary-weighted DACs. It can be shown that for a binary-weighted DAC, the INL at each digital code follows the same normal distribution and has the same variance as shown in (2.6a), while the DNL have large variance at the major carries. The worst-case DNL occurs at the mid-code transition with the maximum standard deviation equal to

$$\max(\sigma_{DNL})|_{binary} = \sigma\sqrt{N - \frac{1}{N}} \tag{2.16}$$

which is two times larger than the maximum standard deviation of INL (see (2.7)). The INLs and DNLs at each digital code can also be expressed in vectors and they follow a similar multi-variant normal distribution as shown in (2.8)~(2.15) except that their covariance matrices ($\vec{\mathbf{B}}$) will be different from that shown in (2.9).

2.3 YIELD ESTIMATION

To meet a given INL specification, for example, within ±A LSB, the magnitude of the INL at any digital code must be no larger than A LSB. Therefore, the INL yield of a normalized DAC (σ=1) can be expressed as

$$\begin{aligned} \Phi(n,A) &= P(|INL(D)| \le A, D = 1,2,\cdots,N-1) \\ &= \int_{-A}^{A}\cdots\int_{-A}^{A} f_{INL}(x_1,x_2,\cdots,x_{N-1})dx_1dx_2\cdots dx_{N-1} \end{aligned} \tag{2.17}$$

where n is the bits of resolution and $N=2^n-1$. If the relative standard deviation of each unit current source is σ, the INL yield of the DAC is equal to

$$Y_{INL} = \Phi\left(n, \frac{A}{\sigma}\right) \tag{2.18}$$

The above expression shows the relationship between the minimum required standard deviation of unit current sources (σ), the bits of resolution (n), the INL (or DNL) specification (A) and the corresponding yield (Y). For the convenience of description later, this expression is rewritten in the form shown in (2.19), where Ψ is an inverse of the two-variable function Φ. With this inverse, the variable A/σ is expressed as a function of n and Y.

$$\frac{A}{\sigma} = \Psi(n, Y_{INL}) \tag{2.19a}$$

or

$$\sigma = \frac{A}{\Psi(n, Y_{INL})} \tag{2.19b}$$

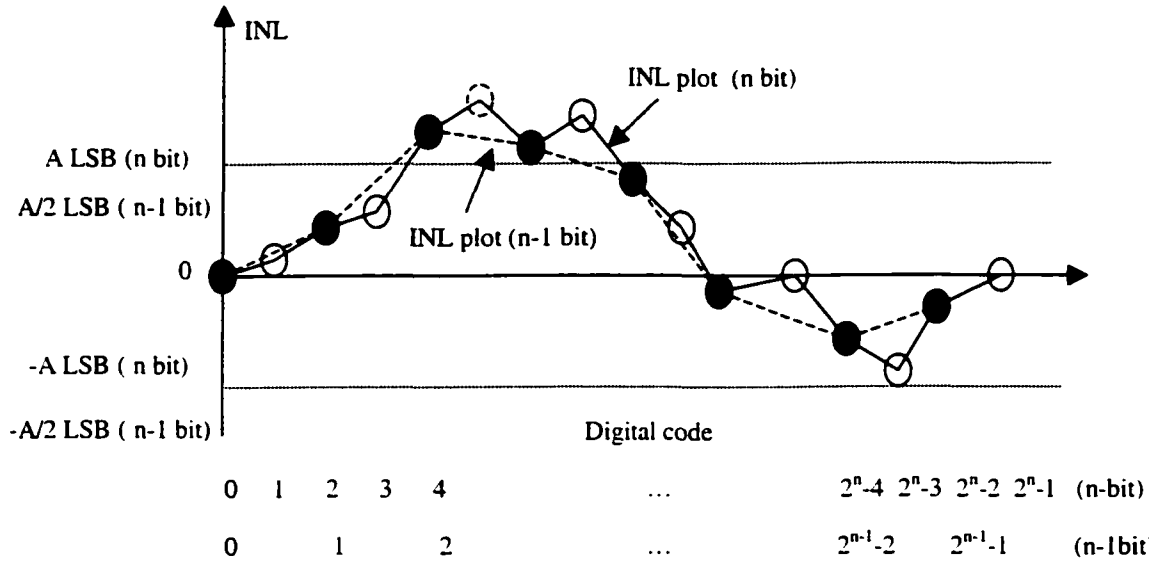
Although the above derivation is for INL yield (Y_{INL}) estimation, similar expressions can also be applied to DNL yield (Y_{DNL}) except that the joint probability density function used in (2.17) will be different.

No one has reported a method of accurately calculating the integral of the joint density function of a multiple-dimensional Gaussion distribution shown in (2.17) without resorting to numerical methods. It appears that approximations have to be made to derive a simple expression. Lakshimikumar has presented two rough bounds for INL yield estimation in binary-weighted DACs [40],[42]. One is overly pessimistic ignoring the strong correlation between different analog outputs and the other is rather optimistic only considering the contribution of two mid-scale codes. The limitations resulting from these assumptions were clearly shown in [41],[43]. Another formula proposed by Bosch for INL yield estimation is based on a nonstandard INL definition and a questionable statistical assumption, where each current output is compared to the ideal value without correcting for the gain error [43]. The

same problem has been found in the simulation results of some recently reported current-steering DAC designs, e.g. [13],[19] and [44]. As a result, the minimum required matching accuracy chosen in these DACs turned out to be significantly overestimated so that the area cost was much more than necessary and thus segmentations made in the DACs are not as good as they could be. In this chapter, a simple but accurate approximation of (2.19) will be given and it can be used for both INL and DNL yield predictions in thermometer-decoded and binary-weighted DACs through the following observations:

A Thermometer-decoded DACs

In thermometer-decoded DACs, the covariance matrix of INL given in (2.9) shows that substantial correlation exists between the INLs at different digital codes. The closer the two codes are, the stronger the correlation is. Figure 2.1 shows a typical INL plot of an n-bit DAC versus the digital codes. It can be seen that the INL of two adjacent codes are very similar. Their difference, as we mentioned before, is approximately equal to ε_D, the relative deviation of the single unit current source I_D, which is negligibly small. As illustrated in Figure 2.1, picking the odd samples (solid circles) in an n-bit INL plot results in an n-1-bit plot (dashed lines) that has almost the same profile as the original plot (solid line), thus the two INL plots must have almost the same probability falling in [-A, A] range. Here, the unit of A (LSB) is referring to n-bit resolution and equivalent to A/2 LSB for n-1 bit resolution. Therefore, the n-bit DAC associated with the original INL plot and the n-1 bit DAC associated with the new INL plot have almost the same probability or yield to achieve INL≤A and INL≤A/2 respectively. Of course, as part of the original INL plot, the new plot



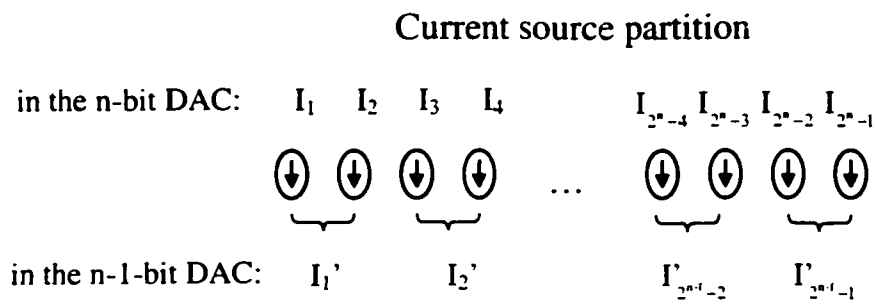


Figure 2.1 (a) INL plots of an n-bit DAC and an n-1-bit DAC obtained by setting the LSB of the n-bit DAC to 0 (b) Current source partitions in the two DACs

will have less chance to go beyond the range [-A, A] and hence the n-1-bit DAC has a little higher yield than its n-bit counterpart. However, these minor differences can be neglected and will diminish as σ decreases. The n-1-bit DAC (obtained from the original DAC by setting the LSB to 0) is actually formed by combining each two adjacent unit current sources in the original n-bit DAC. Thus the jth unit current source in the new DAC, denoted as I_j'($1 \le j \le 2^{n-1}-1$), is equal to

$$I_j' = I_{2j-1} + I_{2j} = 2I \cdot \left(1 + \frac{\varepsilon_{2j-1} + \varepsilon_{2j}}{2}\right) \tag{2.20}$$

and its relative standard deviation is $\sigma' = \sigma/\sqrt{2}$. In summary, in the n-1 bit DAC, $\sigma' = \sigma/\sqrt{2}$, $A' = A/2$, $Y' = Y$. Putting these values into (2.19), we can conclude that

$$\Psi(n, Y_{INL}) = \sqrt{2}\Psi(n-1, Y_{INL}) \tag{2.21}$$

It implies that the reduced number of bits results in an increase in the minimum required standard deviation σ by a factor of $\sqrt{2}$. As we mentioned before, this argument is true when n is large and σ is small. Assume (2.21) is still reasonably accurate when the number of bits is reduced to n_x, and assume we have already known the characteristics of function Ψ for an n_x-bit DAC. For example if n_x=8, we can certainly run Monte Carlo simulations to obtain the function Ψ for a 8-bit DAC. Then for an n-bit DAC, when $n \ge n_x$, the function $\Psi(n, Y_{INL})$ can be easily obtained by repeating the derivation in (2.21) by n- n_x times, that is

$$\Psi(n, Y_{INL}) = \left(\sqrt{2}\right)^{n-n_x} \Psi(n_x, Y_{INL}) \tag{2.22}$$

where $\Psi(n_x, Y_{INL})$ is only a function of Y_{INL} and independent of n. With this in mind, we can rewrite (2.22) as

$$\Psi(n, Y_{INL}) \approx (\sqrt{2})^n / Z(Y_{INL}) \tag{2.23}$$

where $$Z(Y_{INL}) = (\sqrt{2})^{n_x} / \Psi(n_x, Y_{INL})$$

Therefore from (2.19b), the minimum required standard deviation of the unit current sources can be expressed as

$$\sigma = \frac{A}{(\sqrt{2})^n} Z(Y_{INL}) \tag{2.24}$$

Since $Z(Y_{INL})$ depends only on the yield requirement, it can be tabulated for use with arbitrary n if n is reasonably large and the σ obtained is small. Figure 2.2 shows the plot of Z(Y), which was obtained through Monte Carlo simulations assuming A=0.5LSB and n= 8.10 and 12 bit respectively. The similarity of the three curves supports the observation that Z(Y) has little dependence on the number of bits of resolution, n, for most practically used

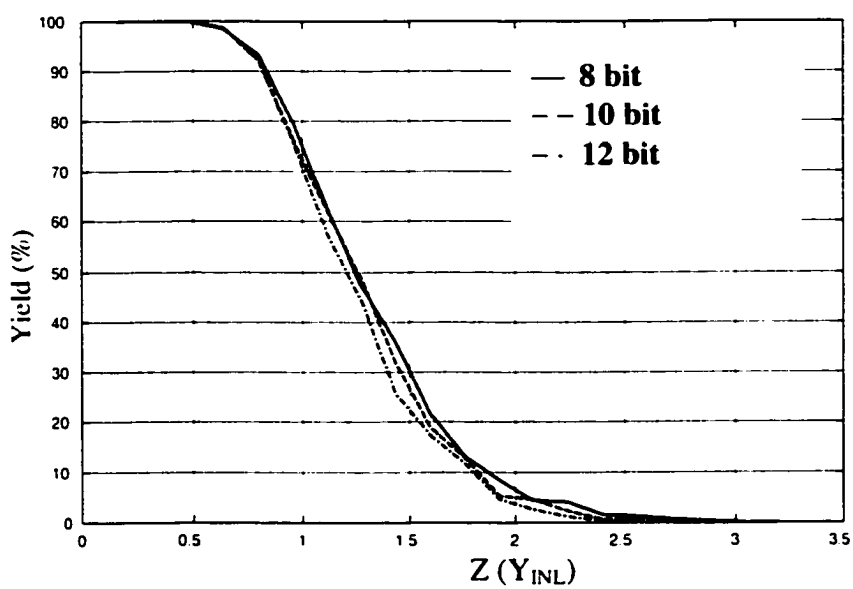


Figure 2.2 Z(Y) for INL yield of Thermometer-decoded DACs

DACs. Therefore, this figure applies for any number of bits, with which we can predict the current source matching accuracy required for a given INL specification and a given yield specification. For example, for a 14-bit DAC, to achieve an INL≤0.5LSB with a 99% INL yield, the relative standard deviation of unit current source has to be less than $0.5 \cdot Z_{INL}(99\%)/\left(\sqrt{2}\right)^{14} \approx 0.22\%$. As mentioned before, the formula given in (2.24) is more accurate for high-accuracy DACs since their unit current sources have smaller deviations.

Figure 2.2 also shows that in the region close to 100% yield level, the yield is not very sensitive to Z(Y) and hence the matching accuracy σ. Following that is a steep region where the yield degrades significantly when σ only increases by a small amount. This is more serious for high-resolution DACs since $Z(Y) = (\sigma/A) \cdot \left(\sqrt{2}\right)^n$. Therefore, in a practical design, some margin should be given to σ in order to avoid large degradation of yield due to run-to-run variations. However, being too conservative is also not preferred because small σ means large area. Besides, when σ is smaller than a certain value, the improvement of the yield becomes insignificant.

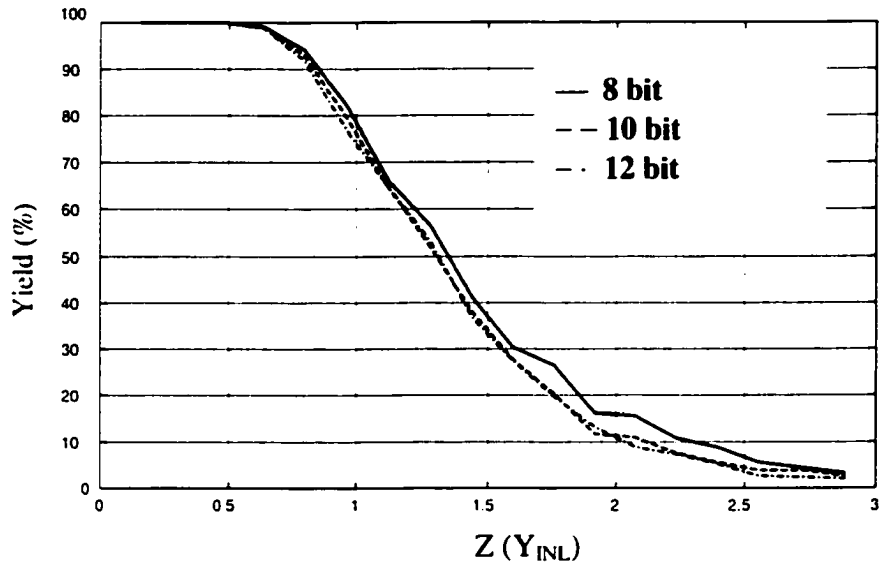
B Bineary-weighted DACs

Similar results can be obtained for binary–weighted DACs as well. It is well known that the nonlinearities of binary-weighted DAC are all associated with the major carries. The severity of the problem is proportional to the weight of the bit [5][44]. Therefore, the linearity of the DAC won't change much when getting rid of the least significant bit (bit0). In another words, the resulting n-1 bit DAC has similar INL and DNL plots as the original DAC. The partition of the $N=2^n-1$ unit current sources in the two DACs are shown as follows:

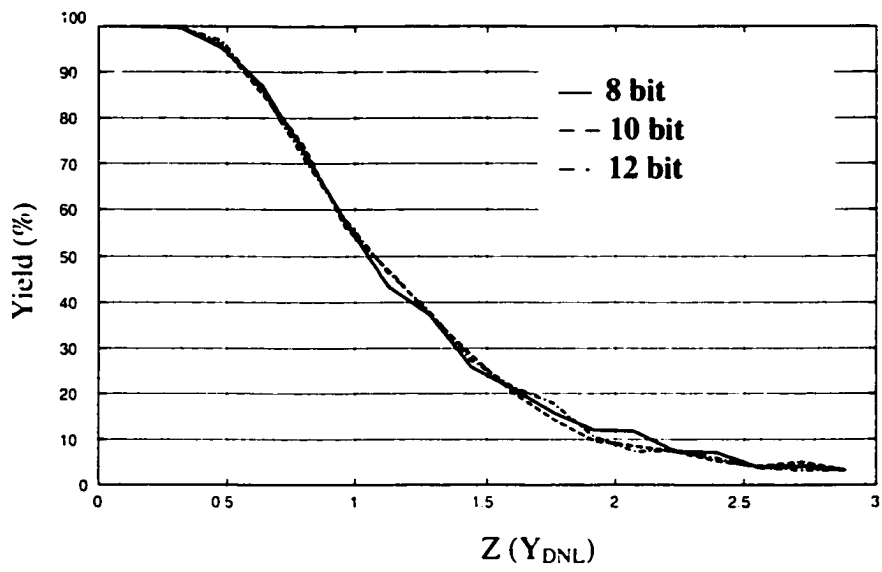
n-bit DAC	n-1 bit DAC
bit0→ I_1	
bit1→ I_2+ I_3	bit0→ $I_1'=(I_2+I_3)$
bit2→ I_4+ I_5 +I_6+ I_7	bit1→ I_2'+ $I_3'=(I_4+I_5)+(I_6+I_7)$
...	...

It is shown that the jth unit current source in the new DAC, denoted as I_j', $(1 \le j \le 2^{n-1}-1)$, consists of I_{2j} and I_{2j+1}, and its relative standard deviation $\sigma' = \sigma/\sqrt{2}$. Therefore, formula (2.21) and (2.24) are also valid for the INL and DNL yield of binary-weighted DAC except that Z(Y) are different. Figure 2.3 (a) & (b) show the plots of Z(Y) for INL yield and DNL yield respectively. It is observed that with the same matching accuracy and the same yield, the DNL of a binary-weighted DAC is larger than its INL. This observation disagrees with the conclusions drawn in [44] that "if the matching of the D/A chip has been designed to achieve the INL yield specification, the DNL specification will automatically be achieved within the same yield requirement even in the extreme case of a full binary topology". We have proved that this statement is not true. Binary structure does suffer more DNL than INL, and segmented architecture provides a better solution to reduce DNL and relax the matching requirements. However, notice that the plots in Fig2.3 (a) are nearly the same as those shown in Figure 2.2, which implies that segmentation helps little for reducing INL.

With these plots of Z(Y) (Figure 2.2 and Figure 2.3) and formula (2.24), the minimum required matching accuracy and a good segmentation can be easily estimated. For example, as we calculated before, for a 14-bit segmented DAC, to achieve INL≤0.5LSB and 99% INL yield, the relative standard deviation of unit current source σ has to be less than 0.22%. If the segmentation is n1+n2, i.e., the n1-bit MSBs are thermometer decoded and the



(a)



(b)

Figure 2.3 Z(Y) for (a) INL and (b) DNL yield of binary-weighted DACs

and the n2-bit LSBs are binary weighted. The DNL of this segmented DAC is similar as the DNL of a n2+1-bit binary-weighted DAC. Therefore, in (2.24), n=n2+1, σ=0.22%, A=0.5LSB, based on the $Z(Y_{DNL}=99\%)$ obtained from Figure 2.3(b), n2 must be less than 11 bits. Otherwise, smaller σ or larger current sources have to be used. These results (σ=0.22% and n2<11) can be verified through Monte Carlo simulations, which show that when n2=10, the yield for both INL and DNL less than 0.5LSB is 99.8%.

2.3 CONCLUSIONS

The random variation of unit current sources is a major factor impacting the linearity of a current-steering DAC. Assuming the current variations are independent and normal distributed, the INL and DNL at each digital code of the DAC also follow normal distributions, however the strong correlation between the current outputs of each digital code results in a very complex statistic expression of yield. Neither this expression nor the widely used Monte Carlo simulations can clearly reveal the relations between the DAC nonlinearity (INL and DNL), the yield, the current matching accuracy and the number of bits of resolution. To gain insight and facilitate optimization during DAC design, simple but accurate formulas were derived in this chapter for both thermometer-decoded and binary-weighted DACs. It is shown that the INL and DNL are proportional to the relative standard deviation of unit current source (σ). To achieve the same linearity and yield, σ must be reduced by $\sqrt{2}$ for each extra bit of resolution, thus the total gate area of the DAC array will increase by a factor of 4. It is also shown that thermometer-decoded DACs have similar INL yield as their binary-weighted counterpart, but very little DNL, while the binary–weighted topology suffers more severe DNL than INL. Segmented architecture turns out to be a good compromise of both structures, and optimal segmentation can be easily achieved using the formulas provided in this chapter.

CHAPTER 3

SWITCHING SEQUENCE OPTIMIZATION FOR GRADIENT ERROR COMPENSATION[2]

3.1 INTRODUCTION

Segmented current-steering DAC architectures are widely used in high conversion rate and high-accuracy digital-to-analog converters. The static performance of a segmented DAC is strongly dependent on the linearity of the unary array driven by the MSBs. Therefore, this chapter only focuses on the linearity of the unary arrays that are thermometer decoded. As we mentioned in Chapter 2, when only random errors are considered, for each extra bit of DAC accuracy, the active area of the unary array increases by more than a factor of four. In high-accuracy DACs, this results in arrays in the range of $1mm^2$. The gradient errors in these arrays can become very significant and must be correctly compensated.

Optimizing switching schemes can reduce the nonlinearity due to gradient errors. This potential has been seen in many current-steering DAC designs [5],[12]-[14],[17],[19],[23]-[25],[27],[28],[46]-[49]. A switching scheme is actually a layout technique. In a current-steering DAC, the switching scheme determines the interconnection between the outputs of the thermometer decoder/latch and the control terminals of the switches in the current matrix. As an example, an 8-bit thermometer decoded current-steering DAC is shown in Figure 3.1. The unary array contains 256 current sources that are ideally identical. The switching scheme determines the order the current sources are switched on as the digital code increases from "1" to 255. The current sources are numbered 1, 2, ..., 255 in the order they are switched on. The unused current source is a dummy current source.

---

[2] © 2002 IEEE. Reprinted, with permission, from reference[45] with minor modification.

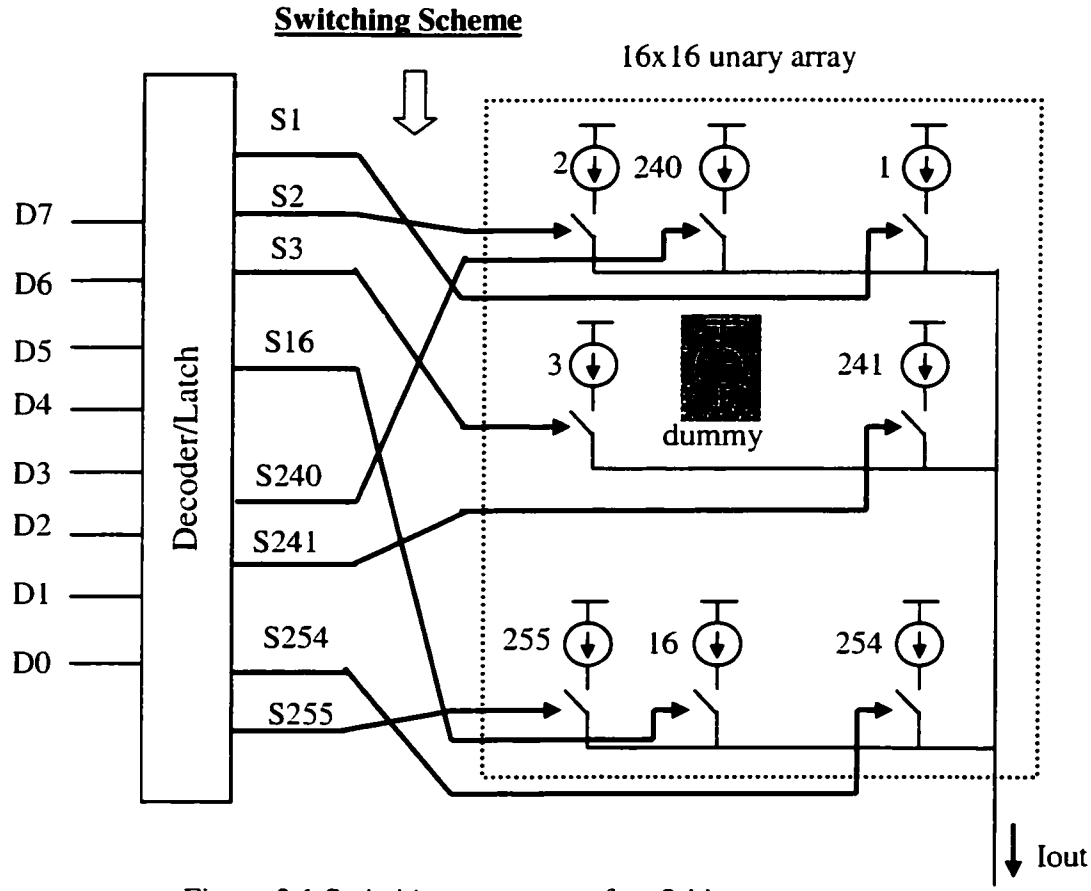


Figure 3.1 Switching sequence of an 8-bit unary array

Instead of inserting a dummy current source there, that area is often used for the biasing circuits. In a segmented DAC, this "dummy" area of the unary array can be used for the binary weighted array. For the unary array in Figure 3.1, there are totally 256! possible switching sequences.

Several switching schemes and sequences have been heuristically derived in literature for gradient error compensation in unary arrays of DACs [5],[12],[14],[17],[19],[23]-[25],[27],[28],[46],[49]. However, no real analytical treatment has been attempted to verify whether these switching schemes and sequences are sufficient to compensate for the gradient errors and the issue whether better solutions exist has not been addressed. A general

approach is necessary to find optimal or near optimal switching sequences under any given type of gradient condition.

An absolute lower bound of INL through optimizing the switching sequence is established in this paper. It will be shown that the conventional switching sequences result in linearity errors that are higher than this lower bound. **Optimal** switching sequences that meet the lower bound are presented for one-dimensional arrays with linear gradient errors. A general approach is developed to find optimal or near optimal switching sequences for any given type of gradients.

Even though current-steering DACs are used as examples here, a similar analysis can be easily applied to other types of thermometer decoded DACs such as capacitor array DACs where charge rather than current is used. Even for resistor string DACs, where the resistor strings are often laid out in several segments, (for example, a 10-bit resistor string containing 1023 resistors are laid out in 32 columns, each column, except one, containing 32 resistors), the order to interconnect the segments (or columns) can also be optimized using the approach described in this chapter.

Before switching schemes are discussed, the linearity errors (INL and DNL) of thermometer decoded DACs are formulized in Section 3.2. This formalization shows a strong dependence of INL on switching sequences. In Section 3.3, typical gradient error distributions are illustrated and normalized. The conventional switching schemes are reviewed in Section 3.4. In Section 3.5, an absolute lower bound of the INL for arbitrary switching sequences is derived and optimal sequences that meet this lower bound are given for one-dimensional linear error arrays. The idea is then expanded to any type of error arrays including two-dimensional (2-D) arrays and a rapidly converging algorithm--the INL bounded algorithm--is developed to find optimal or near optimal switching sequences for any

given type of gradients. Simulation results are given in Section 3.6 to compare different switching sequences.

3.2 LINEARITY ERRORS

To observe how gradient errors affect the INL and DNL of an n-bit unary array, we express the actual current provided by unit current source j ($1 \le j \le 2^n-1$) as

$$I_j = \bar{I} \cdot (1 + \varepsilon_j) \tag{3.1}$$

where $\bar{I}$ is the average current of the 2^n unit current sources in the array and ε_j is the relative deviation of I_j from $\bar{I}$. Hence, the average value of ε_j ($1 \le j \le 2^n-1$) is equal to zero.

For a real DAC, only current sources 1,2,....,2^n-1 are used while current source 0 is a dummy current source. However, in some simplified DACs to be discussed later, the 2^n current sources in the unary array are all used and the digital input is in the range of [0, 2^n] instead of [0, 2^n -1]. For notation convenience, the linearity errors of thermometer decoded DACs will be formulated in two cases: first for DACs without dummy current source, and second, for DACs with one dummy current source.

3.2.1 DACs Without Dummy Current Source

In a unary array, if all the 2^n current sources are numbered 0,1,...,2^n-I in the order they are switched on, the actual output current for digital code D ($1 \le D \le 2^n$) is given by

$$I(D) = \sum_{j=0}^{D-1} I_j + I(0) \tag{3.2}$$

where I(0) is the offset current. The definition of INL and DNL given by (1.2) in Chapter 1.1.1 is repeated below

$$INL(D)=\frac{I(D)}{\bar{I}}-D \tag{3.3a}$$

$$DNL(D)=\frac{I(D)-I(D-1)}{\bar{I}}-1 \tag{3.3b}$$

It follows from (3.1)-(3.3) that the INL and DNL at digital code D can be expressed as

$$INL(D)=\sum_{j=0}^{D-1}\varepsilon_j \tag{3.4}$$

$$DNL(D)=\varepsilon_{D-1} \tag{3.5}$$

The INL and DNL of the overall DAC are defined as

$$INL_{DAC}=\max_{D=1}^{2^n}(|INL(D)|) \tag{3.6}$$

$$DNL_{DAC}=\max_{D=1}^{2^n}(|DNL(D)|) \tag{3.7}$$

Therefore, both the INL and DNL are independent of the average current $\bar{I}$ and can be determined simply by the relative errors ε_j ($1\le j\le 2^n-1$) of the current sources in the array. It is apparent that thermometer decoded DACs can achieve very low DNL_{DAC}. For each unit current source in the array, 50% variation is good enough to obtain a DNL_{DAC} of 0.5LSB. However, it can be shown that with a poor switching sequence, the INL_{DAC} can be very high when gradient errors are accumulated. Our goal is to minimize INL_{DAC} by optimizing the switching sequence.

3.2.2 DACs With One Dummy Current Source

In this case, only 2^n-1 unit current sources in the unary array are used and numbered 1, 2,...,2^n-1 in the order they are switched on while the dummy current source is numbered 0. The actual output current for digital code D ($1 \le D \le 2^n-1$) is equal to

$$I(D) = \sum_{j=1}^{D} I_j + I(0) \tag{3.2'}$$

Since $\bar{I}$ is defined as the average current of all the 2^n unit current sources in the array, including the dummy source, the expressions of INL and DNL in (3.3) are not valid in this case. Based on (1.1), the INL and DNL at digital code D can be expressed as

$$INL(D) = \frac{I(D) - I(0)}{I(2^n - 1) - I(0)}(2^n - 1) - D \tag{3.3a'}$$

$$DNL(D) = \frac{I(D) - I(D-1)}{I(2^n - 1) - I(0)} \cdot (2^n - 1) - 1 \tag{3.3b'}$$

From (3.1) and (3.2'), (3.3') can be rewritten as

$$INL(D) = \frac{\sum_{j=1}^{D} \varepsilon_j + \frac{D}{2^n - 1}\varepsilon_0}{1 - \frac{\varepsilon_0}{2^n - 1}} \approx \sum_{j=1}^{D} \varepsilon_j \tag{3.4'}$$

$$DNL(D) = \frac{\varepsilon_D + \frac{\varepsilon_0}{2^n - 1}}{1 - \frac{\varepsilon_0}{2^n - 1}} \approx \varepsilon_D \tag{3.5'}$$

The INL and DNL of the overall DAC are

$$INL_{DAC} = \max_{D=1}^{2^n-1} (|INL(D)|) \tag{3.6'}$$

$$DNL_{DAC} = \max_{D=1}^{2^n-1} (|DNL(D)|) \tag{3.7'}$$

Comparing (3.4')-(3.7') with (3.4)-(3.7), the DNL and INL of DACs in both cases are nearly the same. Therefore the "dummy" effect is often neglected. As we mentioned before, the DNL of a thermometer decoded array is quite small while the INL may be quite high when gradient error accumulates. The switching sequence is optimized to minimize the error accumulation in the INL. Before addressing the switch sequence optimization, we will first characterize the error distributions across the unary arrays of the DACs.

3.3 GRADIENT ERRORS

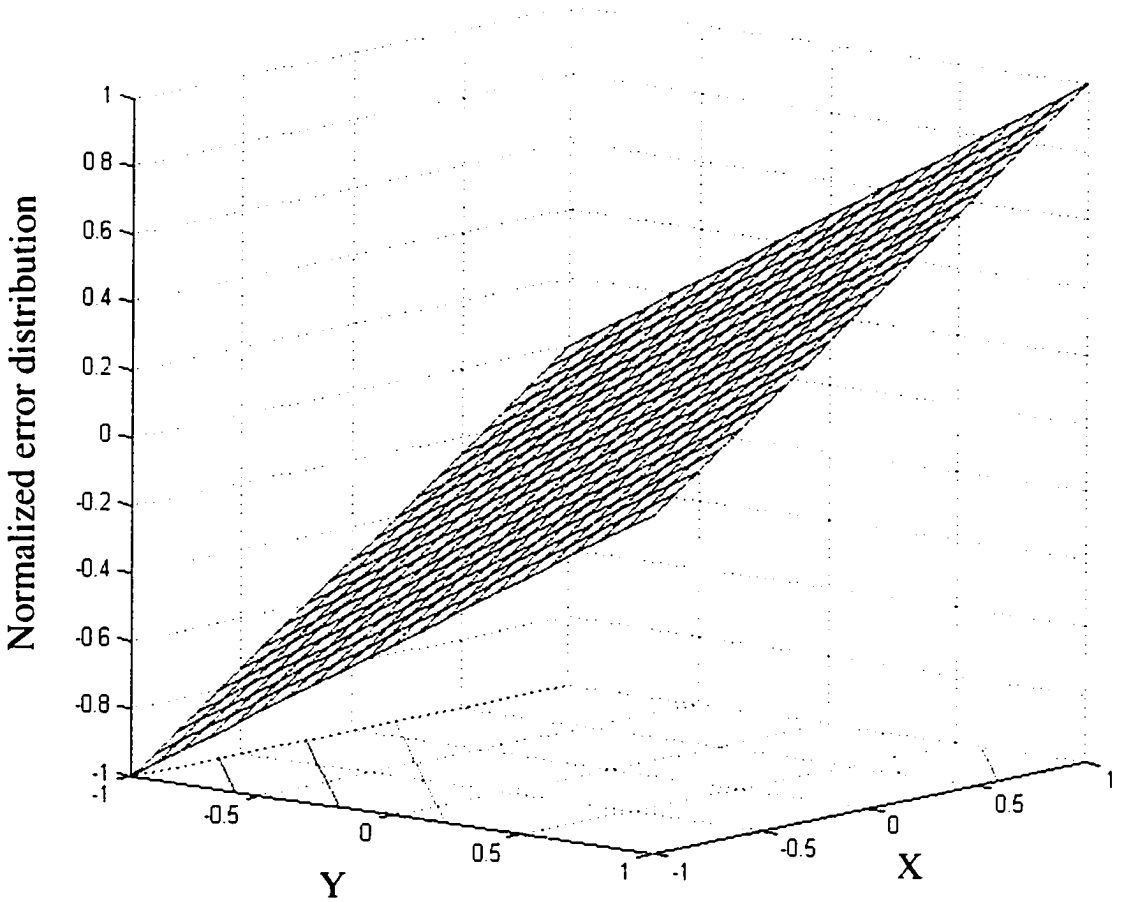
If the errors or mismatch of the unit current sources in a unary array are totally random and uncorrelated, the INL and the yield of a DAC are independent of the switching sequence. However, to overcome random errors, the unary arrays of high resolution DACs generally occupy quite a large area, which results in large distances between current source and hence significant gradient errors [9]. To make the unary array compact, matrix configurations are often used and a square matrix is especially preferred. But even with compact layout, the distances between current sources are still large.

3.3.1 Gradient Error Distributions

Gradient error distribution across a unary matrix can be approximated by a Taylor series expansion around the center of the unary array [12]. The gradient error of the current source located at (x,y) can be expressed as

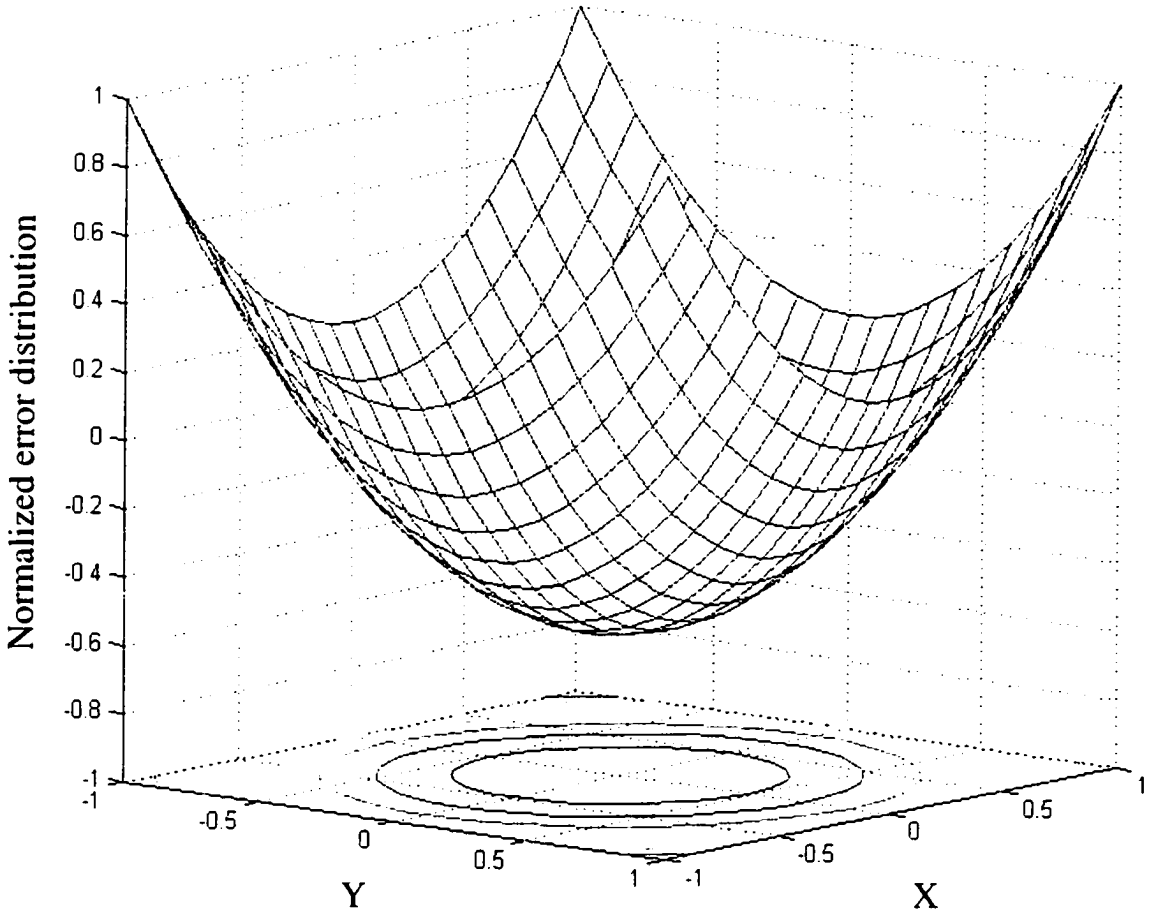
$$\varepsilon(x,y) = a_0 + a_{11}x + a_{12}y + a_{21}x^2 + a_{22}y^2 + a_{23}xy + \cdots \tag{3.8}$$

It is generally assumed that the linear (the first order) and the quadratic (the second order) terms are adequate to model gradient effects [12],[13]. That is, the error distribution is typically linear or quadratic or the superposition of both as illustrated in Figure 3.2. For example, in a current source matrix, the doping and the oxide thickness over the wafer or the voltage drop along the power supply lines have been reported to cause approximately linear



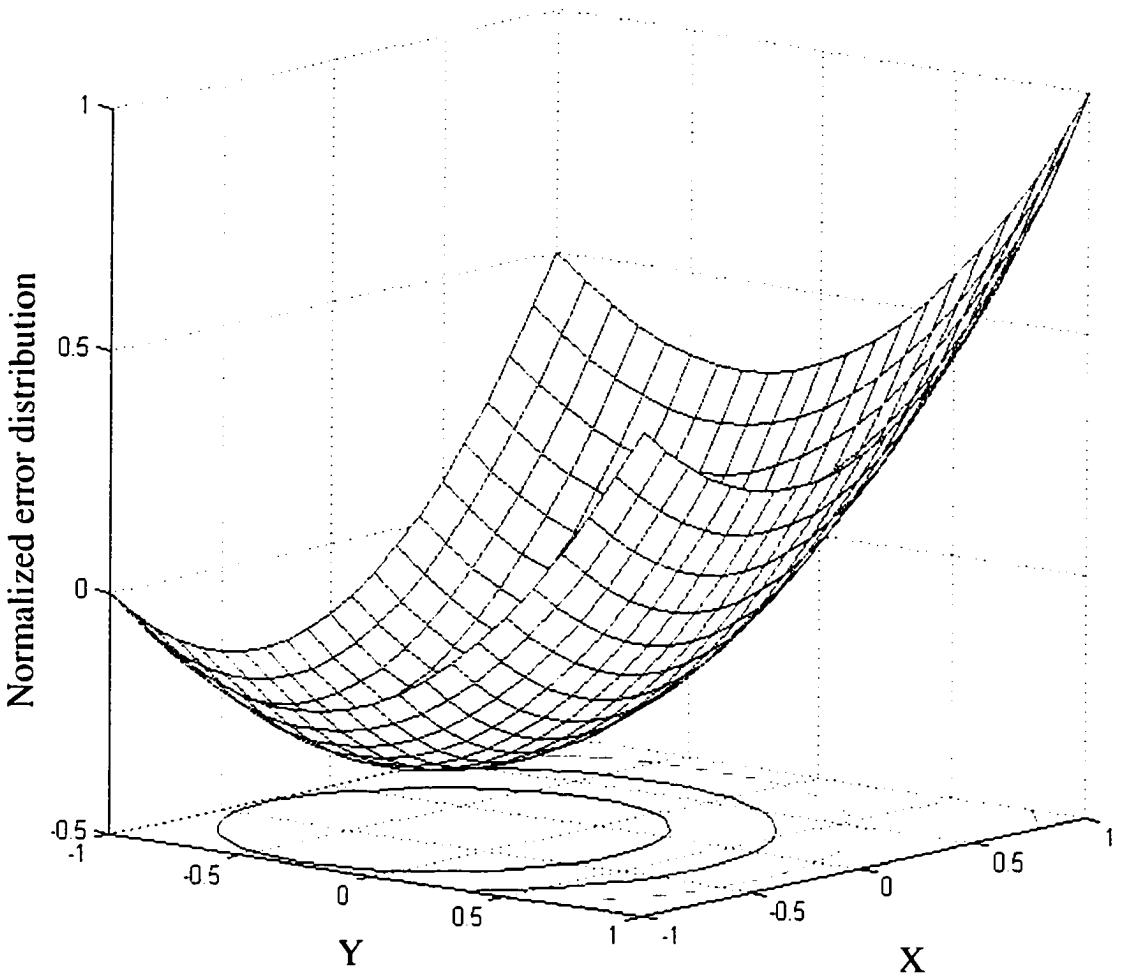
(a) Normalized linear gradient error distribution

Figure 3.2 Models of gradient error distribution (to be continued)



(b) Normalized parabolic gradient error distribution

Figure 3.2 (continued)



(c) Normalized joint gradient error distribution

Figure 3.2 (continued)

gradient errors [12]-[14]. Temperature gradients and die stress may introduce approximately Quadratic errors [16]. The overall systematic error distribution is given by superimposing these error components [14].

Assume in a X×Y matrix layout (X current sources in the x direction and Y current sources in the y direction), the location of each current source is represented by its geometrical center and the spacing between two adjacent sources is Δx in the x direction and Δy in the y direction. Hence the total layout area of the matrix is $(X \cdot \Delta x) \cdot (Y \cdot \Delta y)$. If we use the center of the matrix as the origin as depicted in Figure 3.2, then three typical gradient error distributions in this matrix can be formulized as follows:

A Linear Error Distribution

The linear gradient error for a current source located at (x,y) can be expressed as

$$\varepsilon_l(x,y) = g_l \cdot \cos\theta \cdot x + g_l \cdot \sin\theta \cdot y \tag{3.9}$$

where θ and g_l are the angle and strength of the linear gradient respectively. By definition of (3.1), the average error of the current sources in the matrix is zero. If the linear gradient is due to wafer gradient, since the position of a die on the wafer is unknown, the gradient can occur in any direction, i.e. the angle θ may vary randomly from 0° to 360°.

B Quadratic Error Distribution

It has been observed that the mismatch due to die stress is commonly a symmetrical function of the distance from the die center and that matching sensitivity to die stress is lowest at the die center [16]. As a design rule of thumb, the matching sensitive circuits are suggested to be placed symmetrically at the center of the die. It has also been reported that the stress in the y direction is nearly independent of the x coordinate [16]. In this following, we

assume that the matrix of the DAC is located at the center of the die and the quadratic gradients in both the x and y directions are independent and equal. Therefore, the quadratic gradient error for a current source located at (x,y) can be expressed as

$$\varepsilon_q(x,y) = g_q \cdot (x^2 + y^2) - a_0 \tag{3.10}$$

where g_q is determined dominantly by the die bonding techniques while a_0 is chosen so that the average error of the current sources in the matrix is zero.

C Joint Error Distribution

In this case, the gradient error for a current source located at (x, y) is the superposition of a linear error component and a quadratic error component and can be expressed as

$$\varepsilon(x,y) = \varepsilon_l(x,y) + \varepsilon_q(x,y) \tag{3.11}$$

Notice that (3.11) still keeps the average error equal to zero.

3.3.2 Normalized Error Distribution

From (3.4), it can be seen that if the gradient errors in a unary array all scale by a common factor, then the INL of each digital code, for any switching sequence, also scales by that same factor. In another words, the efficiency of switching sequences is independent of the scaling factor or the strength of the gradient. If a sequence is good for a given error array, it is also good for any scaled version of this error array. Therefore, the comparison between different switching sequences can be made in a normalized gradient error array.

In the following, a square MxM matrix (for an 8-bit array, M is equal to 16) will be used as an example. Since the layout of this matrix is square, the current source spacing in the x direction is equal to that in the y direction, i.e. $\Delta x=\Delta y=\Delta$. The geometric position of this

matrix can then be normalized so that all the current sources are spatially distributed in the interval [-1,1] in both x and y directions as shown in Figure 3.2, i.e.,

$$x, y \in \left\{-1, -1+\frac{2}{M-1}, -1+2\cdot\frac{2}{M-1}, \ldots, 1-\frac{2}{M-1}, 1\right\}$$

Hence, the geometric position in a real matrix can be obtained by multiplying the normalized position by the scaling factor

$$Sp = \frac{(M-1)\cdot\Delta}{2} \tag{3.12}$$

In the normalized MxM matrix, the gradient errors are normalized so that the maximum error magnitude is equal to 1. The denormalization scaling factors will be given under the three typical error distribution conditions.

A Linear Error Distribution

Assume the linear error component is all due to wafer gradient. From die to die, the angle of the wafer gradient θ may vary randomly from 0° to 360°. The maximum possible magnitude of the linear errors occurs when θ=45° and 135°, which is equal to $\sqrt{2}g_1$ (see (3.9)). As shown in Figure 3.2 (a), we normalize this magnitude as 1, i.e. $g_1 = 1/\sqrt{2}$ in (3.9). The overall denormalization scaling factor in this case, including the position scaling factor Sp in (3.12), is

$$S = \sqrt{2}\cdot g_1 \cdot \frac{(M-1)\cdot\Delta}{2} \tag{3.13}$$

B Quadratic Error Distribution

The maximum magnitude of the quadratic errors is equal to $2g_q$-a_0 (see (2.10)). As shown in Figure 3.2 (b), if we set $2g_q$-a_0 =1, then the denormalization scaling factor is

$$S = (2g_q - a_0) \cdot \frac{(M-1) \cdot \Delta}{2} \tag{3.14}$$

C Joint Error Distribution

If the linear error component is due to wafer gradient, the maximum possible magnitude of the overall gradient error occurs when the linear gradient angle θ is 45° or 135°, and is equal to $\sqrt{2}g_l + 2g_q - a_0$. As shown in Figure 3.2 (c), if we normalize this magnitude as 1, then the denormalization scaling factor is

$$S = (\sqrt{2}g_l + 2g_q - a_0) \cdot \frac{(M-1) \cdot \Delta}{2} \tag{3.15}$$

Optimization of switching sequences becomes rather complicated if both linear and quadratic errors are present. We have to consider not only the direction of the linear gradient but also the ratio of the linear component to the quadratic component, which is defined as

$$w = \frac{\sqrt{2}g_l}{2g_q - a_0} \tag{3.16}$$

This ratio can be estimated since the gradients of different types of errors have been experimentally quantified for the technology used.

In summary, the actual INL of the DAC can be obtained by multiplying the INL based on the normalized error array by the denormalization scaling factors given in (3.13)-(3.16).

3.4 CONVENTIONAL SWITCHING SCHEMES

For a given type of gradient formalized as above, the INL of a DAC corresponding to a switching sequence can be calculated based on (3.4) and (3.6). To find the sequence resulting in the lowest INL, one may try to exhaust all possible sequences. However the number of possibilities arises in a factorial fashion with the number of bits and thus becomes incredibly large for DACs with over 4-bit resolution. For a 4-bit unary array which consists of 16 unit current sources, there are $16!=2.1\times10^{13}$ possible sequences, while a 5-bit array has $32!=2.6\times10^{35}$ possibilities.

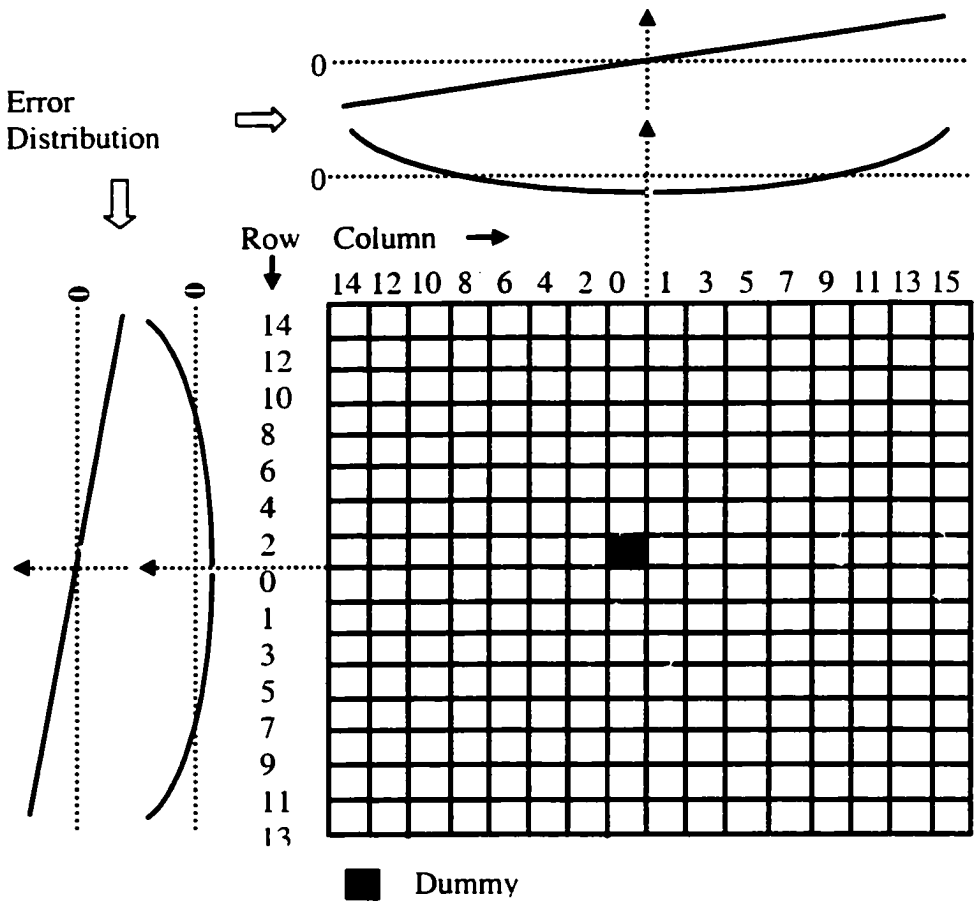
3.4.1 Row-Column Switching Scheme

The well-known row-column switching scheme is commonly used in a heuristic attempt to optimize the switching sequence [5],[13],[14],[17],[47],[48]. In this scheme, the spatial gradient are averaged in two directions as shown in Figure 3.3 and the sequences for row and column selection are optimized independently. The switching optimization problem is thus reduced to a one-dimensional space.

In the 8-bit matrix (16x16) shown in Figure 3.3, "Symmetrical Sequence" [17] is used for row and column selection and the overall sequence for the array is as follows:

(row 0,column 1),(row 0,column 2), ... ,(row 0,column15),
(row 1, column 0),(row 1,column 1),(row 1,column 2), ... ,(row 1,column15),
...
(row15,column 0),(row15,column1),(row15,column 2), ... ,(row15, column15)

The dummy current source is at (row 0, column 0). In this symmetrical sequence, linear errors are cancelled by every two current sources located symmetrically about the center but quadratic errors accumulate.



14 12 10 8 6 4 2 **0** **1** 3 5 7 9 11 13 15

Symmetrical sequence

13 9 5 **1** **0** 4 8 12 14 10 6 **2** **3** 7 11 15

Hierarchical symmetrical sequence (Type A)

14 10 6 **2** **0** 4 10 12 13 9 5 **1** **3** 7 11 15

Hierarchical symmetrical sequence (Type B)

Figure 3.3 Row-column switching scheme

To compensate for both linear and symmetrical (an approximation of quadratic) errors, hierarchical symmetrical sequences were proposed [14]. The hierarchical symmetrical sequences as well as the symmetrical sequence for a 1x16 array are all given in the lower part of Figure 3.3. The hierarchical symmetrical sequence of Type A compensates for symmetrical errors at the first level and compensates for linear errors at the second level. Correspondingly, Type B sequence compensates for linear errors at the first level and compensates for symmetrical errors at the second level.

Even with good switching sequences, the row-column switching schemes are inherently insufficient for two-dimensional gradient error compensation. The gradient error of each unary array current source can be divided into two parts: the column related error and the row related error. As the current sources in one row are turned on successively, and the column related errors are soon compensated due to the optimized column switching sequence. However, the error related to the row accumulates and can not be compensated until the next row is switched on.

The advantage of the row-column scheme is its simplicity for decoder design and layout. Usually, two decoders, one for rows and the other for columns, are used, while each current source cell contains a simple local decoder, a latch/switch driver and a current source.

3.4.2 Hierarchical Switching Scheme

An alternative to the row-column scheme is a two-step hierarchical switching scheme undertaken in the "Q^2 Random Walk" current-steering DAC [12]. As shown in Figure 3.4, the 8-bit (16x16) current matrix is divided into 16 regions (4x4) and each region has 16 current sources (4x4). The switching of regions in [12] (designated as Seq_Q) is in the order of A, B, ..., P and used to compensates for quadratic errors and the switching within each region,

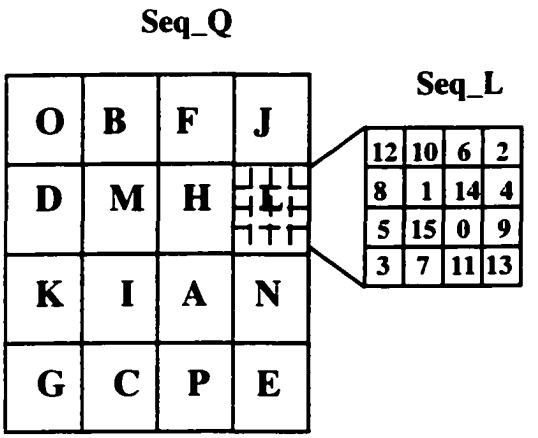


Figure 3.4 Row-column switching scheme

(designated as Seq_L) is in the order of 0,1,2,...,15 and used to compensates for the linear errors. The overall switching sequence is:

(A,0), (B,0), ..., (P,0), (A,1), (B,1), ..., (P,1), ..., (A,15), (B,15), ..., (P,15)

The current source 15 in region P is the dummy source and provides biasing for the circuit. This hierarchical switching scheme allows optimization in 2-D space (even though the optimization is still constrained) with the penalty of very complex routing.

For DAC design, a trade-off has to be made between accuracy and complexity. To further optimize the switching sequences without dramatically increasing the complexity of DACs, we will consider three options:

1) If the row-column switching scheme is used due to its simplicity, optimal sequences are needed for gradient error compensation in one dimensional arrays.

2) If the hierarchical scheme is used, optimal sequences are needed for 2-D gradient error compensation. Reference [12] only gave the switching sequences for linear and

quadratic error compensation in a 4x4 array. The derivation of the two sequences was not presented, and a general approach to find switching sequences for different size arrays was not described either.

3) If we try to find optimal switching sequences through unconstrained optimization over 2-D arrays, it is necessary to derive a good algorithm that can find optimal or near optimal sequences for a given type of gradient without consuming too much time. As a matter of fact, with such an algorithm, the problems in 1) and 2) are also solved.

Other methods for gradient error compensation include providing well-established local biasing for each quadrant of the current matrix [5],[35], splitting each current source into several units located symmetrically in the matrix [12],[13],[19],[23]-[25] and etc. These methods effectively suppress the spatial gradient errors and the linearity of the DACs are thus determined by the residual errors in the local region, which can be further compensated with either the row-column switching scheme or the "random walk" scheme. This implies that combining these methods with the switching schemes and sequences described in this paper may provide more effective gradient error compensation. However, this topic is beyond the scope of this chapter.

3.5 INL BOUNDED SWITCHING SEQUENCES

As in the row-column switching scheme, we will first consider the switching optimization in one-dimensional arrays. As an example, a 1x8 unary array with linear gradient errors is shown in Table 3.1. Row1 of the table are the actual values of the current sources in the array. Their relative errors are given in Row2. Followed are three switching sequences and their corresponding INL which can be easily calculated based on (3.4) and (3.6). It is shown that the sequential sequence results in an INL_{DAC} of 16% due to sever error accumulation. When the symmetrical sequence is used, the linearity error caused by a current

Table 3.1 Switching sequences for a 1x8 linear error array and their corresponding INL

Location → Original array : 4.65 4.75 4.85 4.95 5.05 5.15 5.25 5.35 Error array (%) -7 -5 -3 -1 +1 +3 +5 +7			
Sequences	Errors (%) of the current sources: 0 1 2 3 4 5 6 7	INL (%) of digital code: 1 2 3 4 5 6 7 8	(%)
Sequential seq. 0 1 2 3 4 5 6 7	-7 -5 -3 -1 +1 +3 +5 +7	-7 -12 -15 **-16** -15 -12 -7 0	6
Symmetrical seq. 6 4 2 **0** **1** 3 5 7	-1 +1 -3 +3 -5 +5 -7 +7	+1 0 +3 0 +5 0 **+7** 0	7
A new seq. **1** 5 **3** 7 4 **0** 6 **2**	+3 -7 +7 -3 +1 -5 +5 -1	+3 **-4** +3 0 +1 **-4** +1 0	4

source is canceled when the current source located symmetrically is turned on next. For example, in this example, the first current source to be turned on has −1% error, thus the second to be turned on is the one with +1% error. This sequence results in an INL_{DAC} of 7%, equal to the maximum error magnitude in the error array. The third sequence is new and able to further reduce INL_{DAC} by nearly 50%. In what follows, it will become apparent that the new sequence in this example is an optimal sequence. An optimal switching sequence means for a given gradient, no other switching sequence can achieve an INL_{DAC} less than that achieved by this optimal sequence. Note that the definition of optimality says nothing about uniqueness. For a given type of gradient, there are often several or even many distinct optimal sequences.

3.5.1 An Absolute Lower Bound of INL

To find optimal sequences, we will first determine a lower bound for INL_{DAC}. For a unary array containing N current sources without dummy current source, define the maximum and minimum INL of a certain sequence by the expressions:

$$INL\_max = \max_{D=1}^{N} INL(D) \tag{3.17}$$

$$INL\_min = \min_{D=1}^{N} INL(D) \tag{3.18}$$

Then, based on (3.6), INL_{DAC} can written as

$$INL_{DAC} = \max(INL\_max, -INL\_min) \tag{3.19}$$

As the digital input D increases, the value of INL(D) moves between INL_max and INL_min as illustrated in Figure 3.5. Each step size is determined by the error of the current source currently switched on. The maximum step is equal to the maximum magnitude of the errors (donated as Emax) in the error array. Therefore, the spacing between INL_max and INL_min is no less than Emax. This results in the inequality:

$$INL\_max - INL\_min \geq Emax \tag{3.20}$$

It can be observed from (3.19) that INL_{DAC} is minimized if INL_max and INL_min are symmetrical about 0 as depicted in Figure 3.5. In this case,

$$INL_{DAC} = INL\_max = -INL\_min \tag{3.21}$$

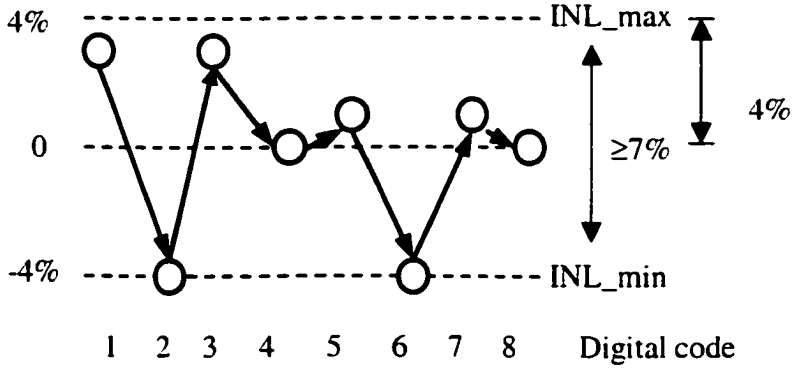


Figure 3.5 Illustration of INL calculation for the new switching sequence in Table 3.1

Returning to (3.20), a lower bound of INL_{DAC} is obtained:

$$INL_{DAC} \geq Emax/2 \tag{3.22}$$

This key inequality establishes an absolute lower bound on the INL of a DAC. It is not dependent upon the type of gradient present and applies to arrays of any dimension.

The formal proof of (3.22) are given as follows: It is well known that if x and y are non-negative real numbers, then $\max(x,y) \geq (x+y)/2$, and they are equal if and only if x=y. Observe that INL_max and –INL_min are non-negative real numbers. It thus follows that

$$INL_{DAC} \geq \frac{INL\_max - INL\_min}{2} \tag{3.23}$$

With (3.20), (3.22) can be obtained and INL_{DAC} is equal to the lower bound Emax/2 if and only if

$$INL\_max=-INL\_min=Emax/2 \tag{3.24}$$

For a given error distribution, Emax/2 is an absolute lower bound of INL_{DAC}. In another words, no switching sequence can results in an INL_{DAC} lower than this lower bound. In the above example, Emax=7%, thus Emax/2=3.5%. Since the resolution of the error array is 1%, the minimum achievable INL_{DAC} is 4%. The new sequence in Table 3.1 meets this lower bound, so it is optimal.

We are now in a position to make the following claim: This new switching sequence given in Table 3.1 is optimal for any 1x8 linear error array independent of both the sign and magnitude of the gradient, because any linear gradient differs from that given in the example only by a constant scaling factor. As mentioned in Section 3.3, the optimality of the sequence will not be impacted by this scaling factor.

3.5.2 Optimal Switching Sequences

The new optimal sequence given in Table 3.1 is not unique. There are several other optimal sequences, two of which number the current sources in the array (from the left to the right) as 2 4 0 7 6 5 3 1 and 3 5 1 7 0 6 2 4 respectively.

We can find optimal sequences by building a tree structure as shown in Figure 3.6. Start with a current source, whose relative error has amplitude equal to or less than the lower bound of INL_{DAC}. In the above example, the lower bound of INL_{DAC} is 4%, so we can start with the current sources that have errors of 3%, 1%, -1% or –3%. They are surrounded with circles in Figure 3.6. If we start with 3%, the INL for digital code "1" is also 3% shown beside the arrow. The next current source is chosen so that the INL for digital code "2" is within [-4%, 4%]. The possible current sources are those with errors within [-4%-3%, 4%-3%]=[-7%, 1%]. As shown in the second row of the figure, -7%, -5%, -3%, -1% and 1% can satisfy this requirement. Likewise, the third current source is chosen so that the INL for digital code "3" is within [-4%, 4%]. The same process is repeated (if possible) until all 8 current sources are selected without repetition, yielding an optimal sequence with which the INL for all digital codes (1-8) are no larger than the lower bound. If the selection is stuck somewhere in the middle, that is, none of the remaining current sources can make the INL meet the lower bound, then the searching fails in this path. We have to go back to the upper level and try another path. Any path successfully going though all 8 levels represents an optimal sequence. For example, in Figure 3.6, the high lighted path: 3%, –7%, 7%, -5%, 5%, -3%, 1%, –1% which corresponds to the sequence 1 3 5 7 6 0 4 2 is another optimal sequence for a 1x8 linear error array.

The same idea can be applied to arrays with different size and different type of gradient, including two-dimensional arrays. In summary, the general form of two algorithms are described as follows:

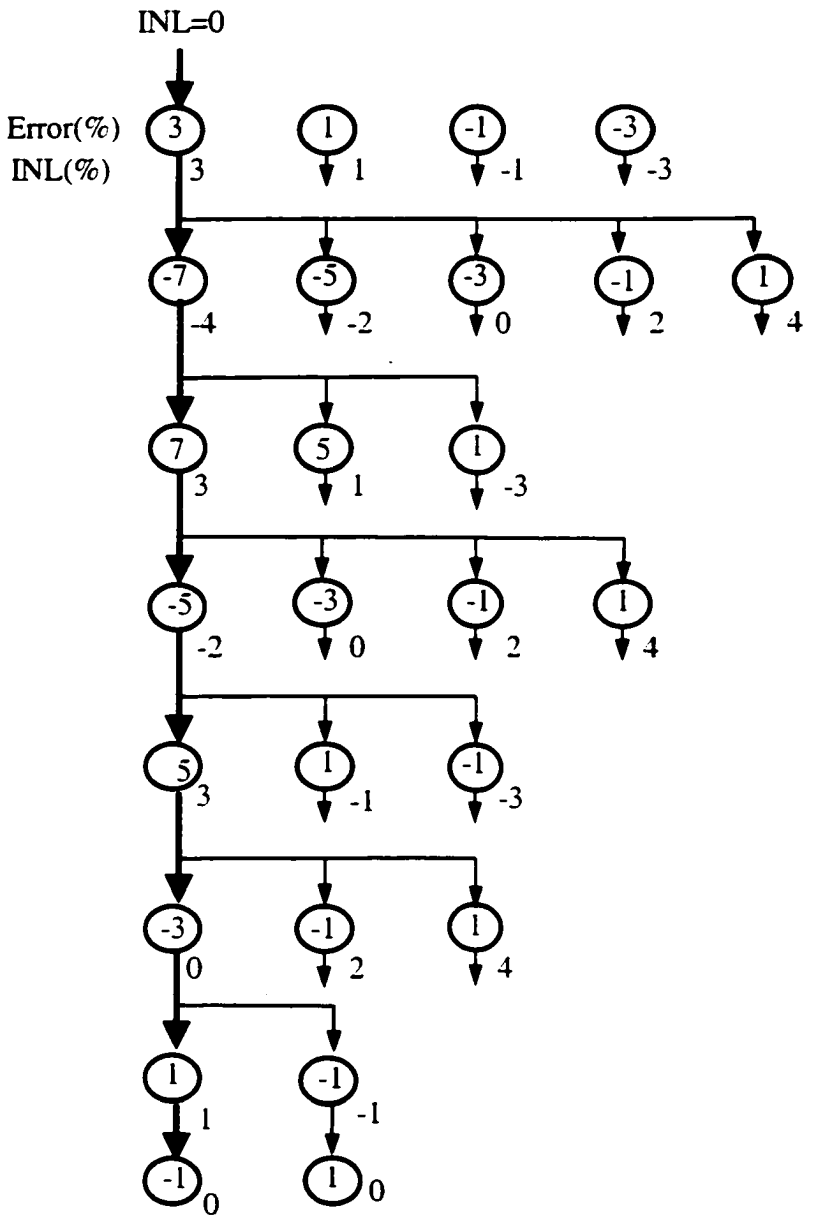


Figure 3.6 Tree structure searching for optimal switching sequences

A *Sort and Group Algorithm*

1) Sort the whole error array in either ascending or descending order. Assuming a unary array consists of 2L current sources, after sorting, we get a new one-dimensional error array labeled as:

$$E_{-L}, E_{-(L-1)}, \ldots, E_{-2}, E_{-1}, E_{1}, E_{2}, \ldots, E_{L-1}, E_{L}$$

Through this step, any 2-D error matrix is reduced to a one-dimensional array. According to (3.22), the absolute lower bound of INL_{DAC} is $\max(|E_L|, |E_{-L}|)/2$.

2) Group the above sorted error array in the same way as those used for one-dimensional linear gradient error arrays, hence the name "Sort and Group (SG)" algorithm. For example, using the grouping method of the new sequence in Table 3.1 results in an error sequence:

$$\underbrace{E_{L/2}, E_{-L}, E_{L}, E_{-L/2}}, \underbrace{E_{L/2-1}, E_{-(L-1)}, E_{L-1}, E_{-(L/2-1)}}, \ldots\ldots \underbrace{E_{1}, E_{-(L/2+1)}, E_{L/2+1}, E_{-1}}$$

The corresponding switching sequence is a SG sequence. Likewise, using the grouping method of the sequence obtained in Figure 3.6 results in another SG sequence.

The optimal sequences for one-dimensional linear error arrays with 8, 16 and 32 current sources are often needed. Some SG sequences for 1x8 arrays have already been given in the above example. The following are two SG sequences, one for 1x16 arrays and the other for 1x32 arrays. They are optimal for linear gradient of any magnitude.

1x16 array: 1 5 9 13, 3 7 11 15, 12 8 4 0, 14 10 6 2

1x32 array: 1 5 9 13 17 21 25 29, 3 7 11 15 19 23 27 11,

28 24 20 16 12 8 4 0, 30 26 22 18 14 10 6 2

Although SG sequences are optimal for one-dimensional linear error arrays, their efficiency for other types of gradient has not been theoretically investigated. Simulation results show that they may not be the optima if quadratic gradients present. Even for two-

dimensional linear error arrays, the optimality of SG sequences can not be guaranteed, because after sorting, the 2-D linear error matrix turns into a one-dimensional error array which is not simply linear. In these cases, some of the SG sequences may perform better than the others. However, for typical gradient distributions as given in Section 3.3, simulation results show that the SG sequences can usually achieve better linearity than the conventional switching sequences.

B INL Bounded Algorithm

A more general approach that may allow further reducing INL_{DAC} is to build a tree as shown in Figure 3.6. Notice that the lower bound given in (3.22) may be an over-optimistic estimation. It is possible that a switching sequence meeting this absolute lower bound does not exist. A practical approach using the tree of Figure 3.6 is to relax the bound of INL_{DAC}. For example, a value between the absolute lower bound given by (3.22) and the INL_{DAC} achieved by the SG sequences could be established. This relaxed bound enhances the possibility of convergence. The sequence obtained by this algorithm is hence termed as an "INL bounded" sequence. These INL bounded sequences are often optimal or near optimal and can sufficiently compensate for any given type of gradient errors, which will be demonstrated in the next section.

The INL bounded algorithm is a simple algorithm, which has not been optimized to minimize computing time although computation minimization strategies could be explored. We have, however, succeeded in obtaining near optimal switching sequences with small computation times for many examples. Even for a unary array with 8-bit resolution, the simple INL bounded algorithm converges pretty fast, and the obtained sequences, as they will be shown in the next section, achieve sufficiently low INL_{DAC}.

3.6 SIMULATION RESULTS

To demonstrate the application of the new switching optimization algorithms and compare the INL bounded sequences with the conventional sequences, a 16x16 matrix of an 8-bit thermometer decoded DAC is used as an example. The error distributions across the matrix are normalized in the way given in Section 3.3. Thus, the INL_{DAC} obtained in the following simulations are normalized. They are only used for purpose of comparison, the actual INL_{DAC} would be denormalized as described in Setion 3.3 to reflect the actual gradient effects. The simulations will be done for both row-column and hierarchical switching schemes and the INL bounded sequences will be given under three typical error distribution conditions.

3.6.1 Using Row-Column Switching Scheme

The sequences that compensate for the gradient errors in a 1x16 array serve as the row and column selection sequences. The symmetrical sequence and the hierarchical symmetrical sequence for a 1x16 array have already been given in Figure 3.3. The INL bounded sequence described in the previous section will be derived under three conditions:

A Normalized Linear Gradient Error

We have already obtained optimal switching sequences (SG sequences) for one-dimensional linear error array in Section 3.5. One for a 1x16 array numbers the current sources (from the left to the right) as

1 5 9 13, 3 7 11 15, 12 8 4 0, 14 10 6 2

A SG Sequence (also an INL bounded sequence)

If the linear gradient is due to wafer gradient, assuming the gradient may go through the array in any direction and with equal probability, then the INL of the DAC versus the angle of the gradient and the yield can be simulated and shown in Figure 3.7. Here, only the symmetrical sequence is compared with the SG sequence, because when only linear gradients are present, the hierarchical sequence of Type B has the same performance as the symmetrical sequence while Type A performs poorly compared to Type B. As we expect, the SG sequence results in an INL_{DAC} nearly 1/2 less than that obtained by the symmetrical sequence and the yield for a given INL_{DAC} can be substantially enhanced.

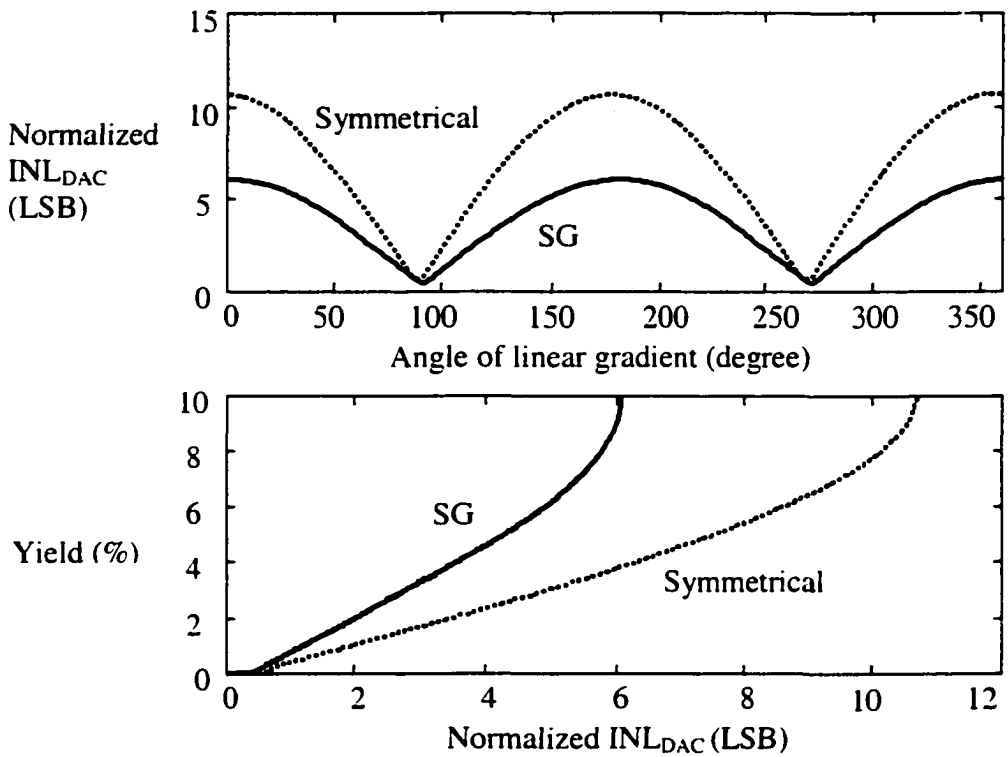


Figure 3.7 INL and yield of the DAC (with linear gradient) using row-column switching scheme

B Normalized Quadratic Gradient Error

We obtained the following INL bounded sequence to compensate for one-dimensional quadratic errors.

7 5 3 1 4 0 11 6 8 14 9 12 2 15 13 10

INL bounded sequence As shown in Table 3.2, the INL bounded sequence results in an INL_{DAC} that is only 1/3 of that attained by the hierarchical symmetrical sequence of Type A. The Type B sequence and the symmetrical sequence are not well-suited for managing quadratic errors and thus are not included in the comparison.

Table 3.2 INL of the DAC (with quadratic gradient) using row-column switching scheme

Sequences	Normaiized INL_{DAC} (LSB)
Hierarchical Symmetrical (Type A)	11.37
INL bounded	4.25

C Normalized Joint Gradient Error (assuming w=1 in (3.16)).

In this case, if the linear gradient is due to wafer gradient so that the angle of the linear gradient (θ) is random, the characteristics of the error distributions in both the row and column directions change with θ. The switching sequence that is optimal for one angle may not be optimal for other angles. Assuming θ varies from 0° to 360° with equal probability, our goal is to find a sequence that is good for most angles or results in a low INL_{DAC} with high yield. In the INL bounded algorithm, ideally the bound of INL_{DAC} should be applied to error arrays resulting from any possible angles. In practice, we apply the bound of INL_{DAC} to the error arrays with typical values of θ, for example, 0°, 45°, 90°, 135° and 180°. Using this approach, we obtained the following INL bounded sequence for a 1x8 array:

12 3 7 0 15 6 8 11 4 13 1 9 2 14 10 5

INL bounded sequence

As shown in Figure 3.8, if 95% yield is required, the INL of the DAC when using the new sequence is only 1/2 of that when hierarchical symmetrical sequences are used.

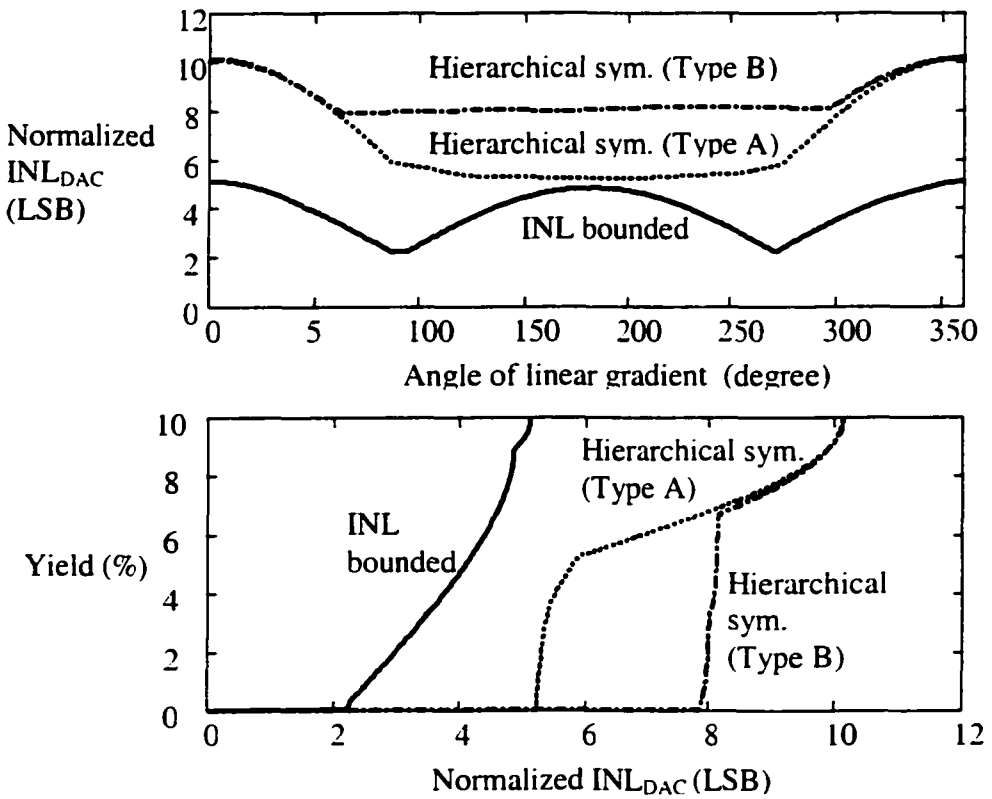


Figure 3.8 INL and yield of the DAC (with joint gradient) using row-column switching scheme

3.6.2 Using Hierarchical Switching Scheme

As in the "random walk" DAC introduced in Section 3.4, the 16x16 array of the 8-bit DAC is divided into 16 (=4x4) regions and each region contains 16 (=4x4) current sources. The switching sequences optimized for a 4x4 matrix control the region selection and the switching within each region. Again, the INL bounded sequences are given under three conditions:

A ***Normalized Linear Gradient Error***

If the linear gradient is due to wafer gradient, a low INL_{DAC} and high yield sequence as shown below can be obtained by applying an INL_{DAC} bound to 4x4 normalized error arrays with typical linear gradient angles.

12	14	6	4
10	8	0	2
1	7	15	9
3	5	13	11

Recall that the Seq_L sequence of the "random walk" scheme (see Figure 3.4) was claimed having the potential to compensate for linear gradient errors. As shown in Figure 3.9, if we use this sequence for both the region selection and the switching within each region, the INL of the DAC varies from 2.3 to 3.4 when the angle of the gradient changes from 0° to 360°. Instead, if the INL bounded sequence given above is used, the INL of the DAC varies between 1.7 and 2.3.

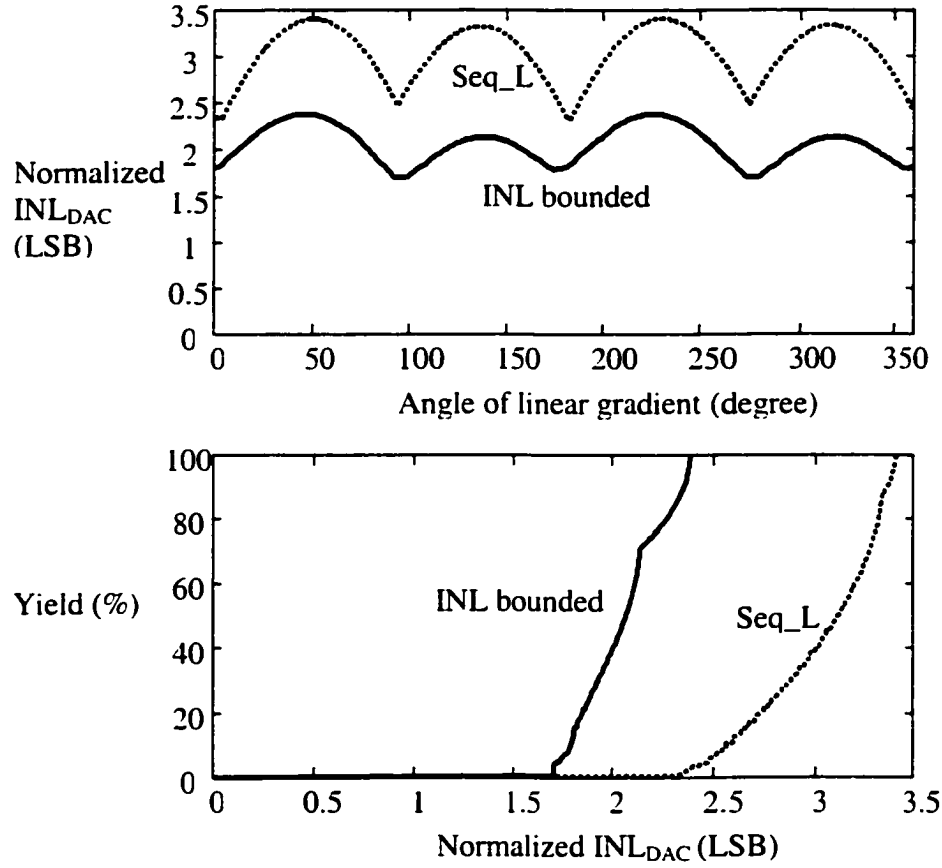


Figure 3.9 INL and yield of the DAC (with linear gradient)
using hierarchical switching scheme

B Normalized Quadratic Gradient Error

In this example, the sequence we obtained using the INL bounded algorithm to compensate for quadratic gradient happens to be the same as the Seq_Q sequence of the "random walk" scheme (see Figure 3.4). Therefore, we still use Seq_Q for the region selection. We can assume the residual gradient within each region is approximately linear. If we use the INL bounded sequence obtained in the above 3.6.2.A section to control the switching within each region, the INL of the DAC is 1.29. Instead, if the Seq_L sequence is used, the INL of the DAC is 1.63. These results are summarized in Table 3.3.

Table 3.3 INL of the DAC (with quadratic gradient)
using hierarchical switching scheme

Sequences	Normalized INL_{DAC} (LSB)
Random walk	1.63
INL bounded	1.29

C Normalized Joint Gradient Error (assuming w=1 in (3.16))

Assume the linear gradient has random directions, the INL bounded sequence for region selection is shown below:

13	0	8	10
2	15	4	5
6	2	9	1
3	11	7	14

We still assume the gradient within each region is approximately linear, so the INL bounded sequence obtained in the above 3.6.2.A section is used for the switching with each region. In Figure 3.10, this switching scheme is compared with the "random walk" scheme. If 95% yield is required, The INL_{DAC} of the "random walk" scheme is 2.0 while the INL_{DAC} of the new scheme is only 1.3.

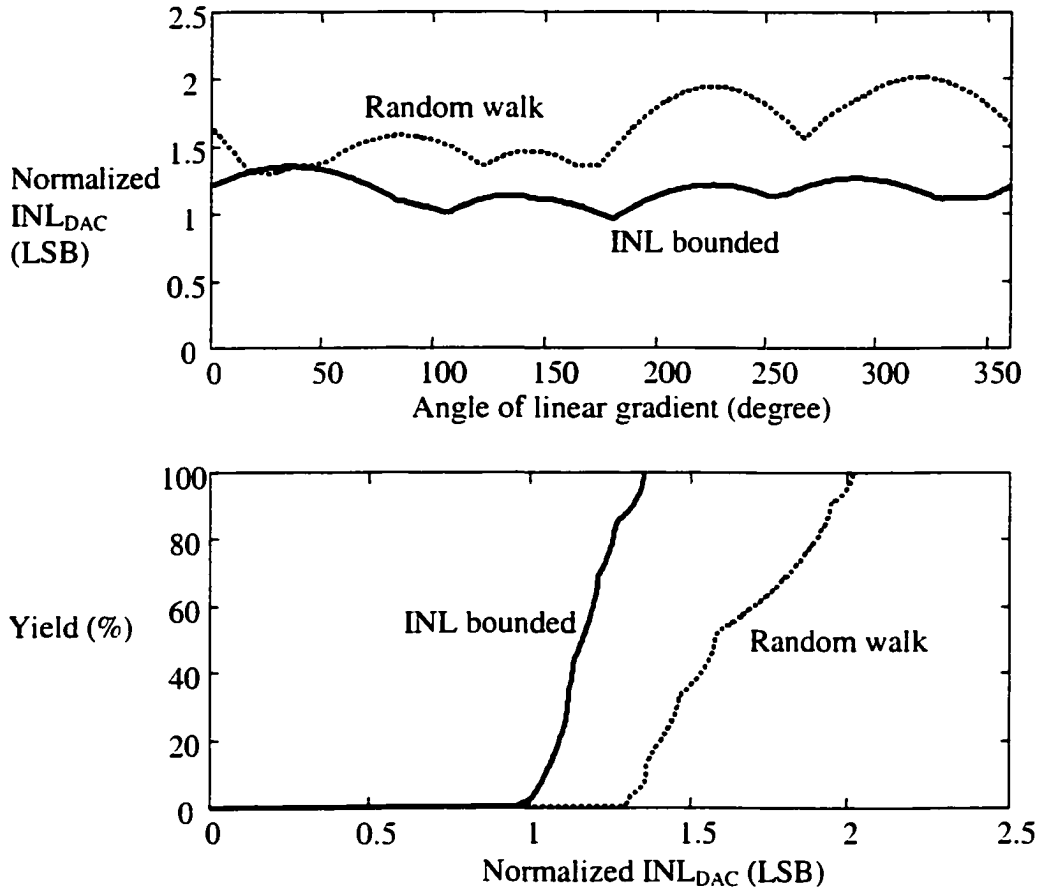


Figure 3.10 INL and yield of the DAC (with joint gradient) using hierarchical switching scheme

Table 3.4 summarizes the performance of the above switching schemes under the three gradient distribution conditions. The hierarchical switching scheme apparently outperforms the row-column schemes. Meanwhile, in both schemes, the INL bounded sequences show significant advantages over the conventional sequences. When row-column switching scheme is used, compared to the symmetrical and hierarchical symmetrical sequences, the INL bounded switching sequences can further reduce the INL of the DAC by approximately 50%. When hierarchical schemes are used, the INL bounded switching

Table 3.4. Performance comparison of different switching schemes under three error distribution conditions

Error distribution	Worst-case INL			
	Row-column switching scheme		Hierarchical switching scheme	
	Conventional	INL bounded	Random walk	INL bounded
Linear	10.7	6.1	3.4	2.4
Quadratic	11.4	4.2	1.6	1.3
Joint (50%linear+ 50%quadratic)	10.1	5.1	2.0	1.4

sequences reduce the INL of the DAC by about 30% compared to the "random walk" sequences.

This example shows that with the Sort and Group (SG) and INL bounded algorithms, optimal or near optimal switching sequence can be obtained when some gradient information, such as the ratio of the linear component to the quadratic component of the gradient, is available. The "Q^2 Random Walk" switching scheme [12] was established based on the good systematic error profile information from a test chip. With this scheme, 14-bit intrinsic accuracy was achieved without trimming or tuning. It has been shown that the two-step hierarchical switching optimization in the " Q^2 random walk" DAC provides much more flexibility for switching optimization than the classical row-column scheme. Unfortunately, the method of determining the switching sequences Seq_Q and Seq_L were not well explained in [12], making it difficult to extend this hierarchical switching scheme to arrays with different size and different types of error profiles. The INL bounded algorithm introduced in this paper provides this flexibility.

3.7 CONCLUSIONS

In this chapter, an absolute lower bound of DAC integral nonlinearity due to gradient effects is established for switching sequence optimization in thermometer decoded DAC arrays. This lower bound is equal to 1/2 of the maximum relative deviation of all the current sources from their average current value. Optimal switching sequences that meet this lower bound were introduced for linear error compensation in one-dimension arrays. A rapidly converging algorithm was developed to provide INL bounded switching sequences for any given type of gradient error condition. Simulation results show that hierarchical switching scheme outperforms the row-column scheme in the presence of linear and/or quadratic gradients. Compared with what is attainable with the best published switching sequences, the INL bounded switching sequences can reduce the linearity errors due to gradient mismatch by up to 50%.

CHAPTER 4

A 1.5V 100MS/s SELF-CALIBRATED DAC

4.1 INTRODUCTION

In the previous chapters, we mainly focused on the static linearity of current-steering DACs. Ad hoc, the static properties only set the best-case performance of a DAC [6]. With the increasing of sampling and/or input signal frequency, the DAC linearity as measured by the SFDR, degrades rapidly because a longer portion of the sampling cycle is occupied by the highly nonlinear switching transients [20],[21]. This transient behavior is strongly dependent on the layout and the process used, and is therefore very hard to predict. Although many techniques have were used in the literature to reduce the transient nonlinearities, they are only effective for one or two error components, sometimes even with the penalty of introducing extra errors.

As we mentioned in Chapter 1.2, the most straightforward and effective method to improve the dynamic performance is to reduce the parasitic capacitance and fast the settling. In this way, even if the switching transition is highly nonlinear, it is short compared to the overall conversion period, leaving the static nonlinearity dominant. Reducing parasitic effects requires small transistors and short interconnections. These requirements, however, conflict with the strategies we previously used for improving the static linearity of DACs. To overcome the random mismatch and gradient errors, the current source arrays were laid out in large dimensions and with complex routing, resulting in large junction and interconnection parasitic capacitance, which severely limit the conversion rate and high-frequency

performance. The serious degradation of SFDR as signal frequency increases can be clearly seen from [12], where an uncalibrated 14-bit intrinsic DAC is reported.

Proper calibration turns out to be a good way to solve this problem. With a small amount of extra circuitry, the area of the current source array can be dramatically reduced while high linearity is still maintained. The decrease of gradient effects also significantly relaxes the requirements on layout and reduces the interconnection capacitance since simple switching schemes can be used

Calibration not only provides the potential to enhance dynamic linearity, but also the potential to overcome technology barriers and achieve guaranteed high static linearity. It was shown that with the matching property of today's technology, 14-bit static linearity is still hard to be achieved without using any trimming or calibration [12]. Calibration also helps to reduce the sensitivities of transistors to process, temperature and aging, therefore ensuring high yields.

As the process feature size keeps shrinking in the favor of digital circuits, so is the supply voltage, which introduce more challenges to high-performance DAC design. For example, in a basic current cell as shown in Figure 1.3(a), since the threshold voltage does not scale down with the same ratio as the supply voltage, the reduction in the effective gate-source voltage of the current source significantly degrades its matching property. To maintain the matching accuracy, a larger current source has to be used, resulting in more severe gradient and parasitic effects.

Low supply voltages also make it impractical to add cascode stages, which is a widely used technique to enhance the output impedance of current sources and improve the DAC dynamic linearity [18]. For a new 0.13μ process and a supply voltage as low as 1.5V, the stacking of the cascode stage further reduces the effective voltage headroom of all the transistors in the current cell and may even make them marginally operational. The

significant reduction of the effective gate-source voltages of the current source severely deteriorates the matching and noise immunity of the current source. In addition, to conduct a reasonable amount of current with low excess bias, the width of all the transistors, including the current source, the cascode stage and the switches become very large thus increasing their parasitic capacitance.

As it will be shown later in this chapter, some other conventional methods used to improve dynamic performance are not suitable for low-voltage designs either. In these cases, calibration becomes inevitable if high resolution is to be maintained at high frequencies.

In this chapter, calibration schemes suitable for high accuracy, high speed and low-voltage applications are investigated. Since most calibration methods that have appeared in the literature are not feasible for very-low-voltage CMOS processes, a new foreground calibration technique is presented in this chapter that effectively calibrates the current source mismatches and has a reduced output impedance requirement. As a result, high linearity and high speed can be achieved with a very small die size and low power dissipation. To demonstrate this calibration technique, a 14-bit current-steering DAC prototype was designed and fabricated in a 0.13μ digital CMOS process. It is the first 14-bit CMOS DAC ever reported that operates with a single1.5V power supply, occupies an active area as small as $0.1mm^2$ and requires only 16.7mW power at 100MHz sampling rate, but still maintains state-of-art linearity and conversion rates.

Although it is desirable to present the proposed DAC strategy in a technology independent way, there is such tight coupling between a semiconductor technology and architectural tradeoffs for a given design that it becomes necessary to consider the specific characteristics of a technology throughout the design process. In this chapter, I will focus specifically on the implementation in a 0.13μ five-metal N-well CMOS process, although I believe the design strategy is applicable in a variety of low-voltage semiconductor processes.

A brief summary of the typical characteristics of MOSFETs in a 0.13μ CMOS process is given in Table 4.1.

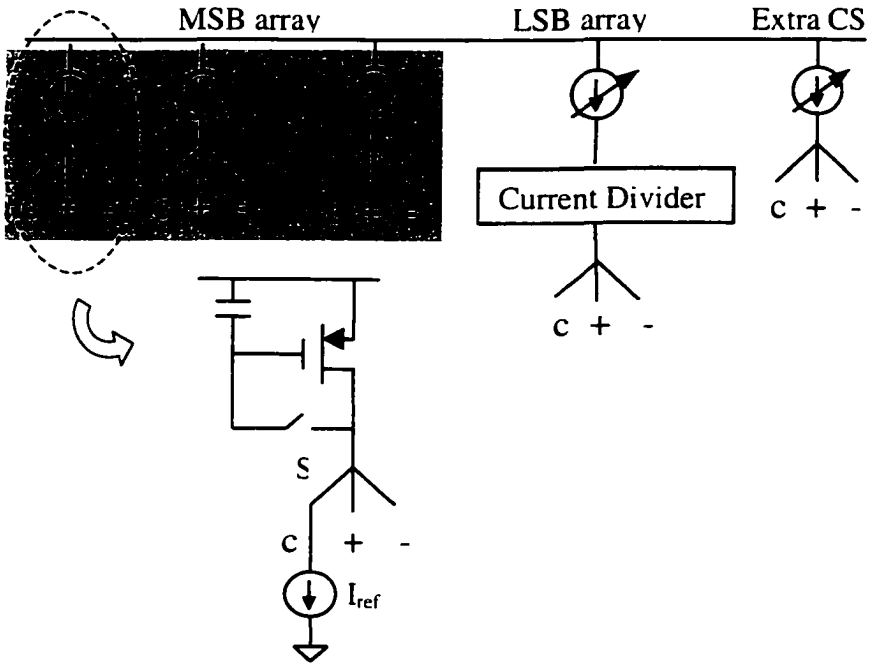
Table 4.1 Typical characteristics of MOSFETs in a 0.13μ CMOS process

	V_t	Tox	Cox	K'=μCox/2
NMOS	0.38V	34 Å	2.5fF/μm^2	250μA/V^2
PMOS	0.45V	35 Å	2.3fF/μm^2	50μA/V^2

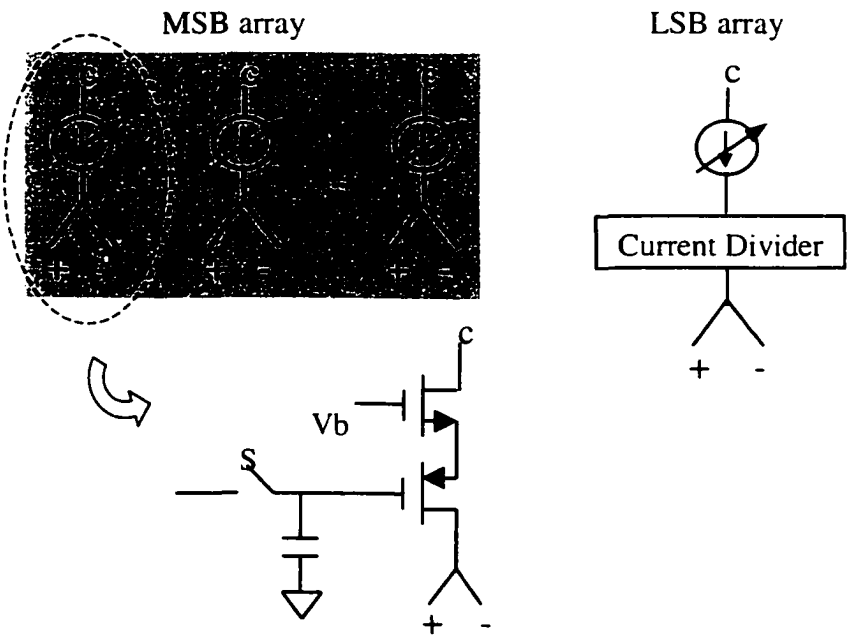
4.2 CALIBRATION SCHEMES

4.2.1 Conventional Calibration Schemes

A widely used background calibration scheme was proposed in [51] where each individual current source in the MSB array as well as the total current of the LSB array (including a dummy current source of 1LSB) are calibrated to be equal to a reference current. The gate-source voltages of the current sources are adjusted accordingly during the calibration and held by their gate capacitors after the calibration. Figure 4.1 (a) shows a conceptual illustration of this scheme where care must be taken to minimize the mismatch of charge injection and clock feedthrough occurring when the calibration finishes and the switch S is open. Due to current leakage, the calibration has to be done frequently to refresh the gate voltage of each calibrated current source. To avoid interrupting the normal data conversion, this calibration is completed in the background. A well-known method for moving the calibration to the background is to use an extra current source as shown in Figure 4.1 (a) that serves as a substitution source when one of the current sources is under calibration. With current sources being switched in-and-out of the current array for calibration, spurs at the



(a)



(b)

Figure 4.1 Conventional background calibration using
(a) An extra current source (b) floating current sources.

calibration frequency may be introduced into the output current [21]. Since the calibration is done sequentially, current sources calibrated earlier in the cycle may lose more current than those calibrated later, resulting in systematic mismatches. Some variations of the scheme have been proposed to optimize the switch network[52],minimize the undesirable spurious signals caused by the calibration and charge leakage[53] and improve the performance at high operating speeds [54]. It is worth mentioning that the LSBs of the DAC are realized by using a current divider underneath a calibrated current source. This multiple stages of stacks is not tolerable for supply voltage as low as 1.5V.

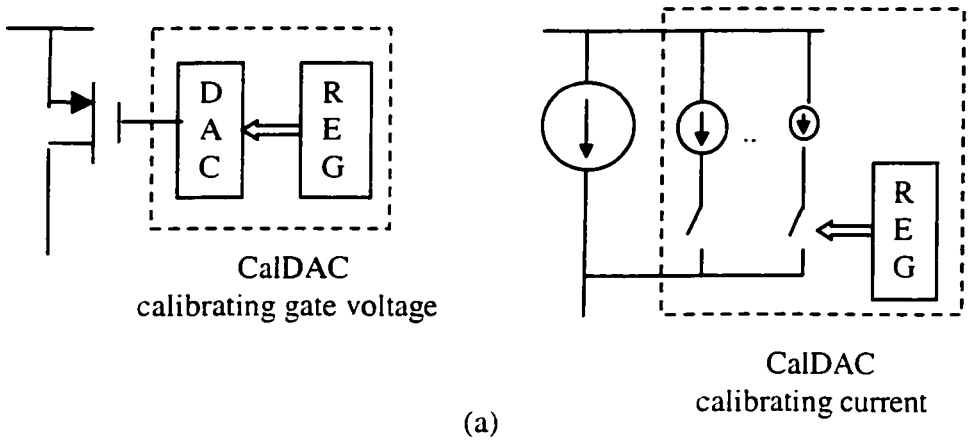
A method to avoid using the extra current source takes advantage of a floating current source or a PMOS-and-NMOS pair as shown in Figure 4.1 (b), where the normal current switching is done on the bottom side of the transistor pair while the calibration is undertaken by sensing the current from the top side of the transistor pair [21]. Therefore, the normal operation and the calibration can be done simultaneously. In this case, the total current of the LSB array (including a dummy current of 1LSB) serves as the reference current and each current source in the MSB array are adjusted to be equal to the reference current. Different versions of this structure are seen in [55] and [56]. A drawback of the transistor pair is the large voltage headroom it requires, which almost doubles that required for a standard current source. Besides, matching has to be considered for both PMOS and NMOS transistors. In this case, two arrays, one consisting of PMOS devices and the other consisting of NMOS devices, are needed. Each of them has to maintain similar matching accuracy, and hence occupy comparable area as a standard current array.

The above approaches are called analog calibrations since the errors are measured and stored as analog signals. Alternatively, the errors can be digitized using a slow but accurate analog-to-digital converter (ADC) and stored in registers or RAMs. During the conversion, these error messages are read out to either adjust the digital inputs or drive a calibration DAC

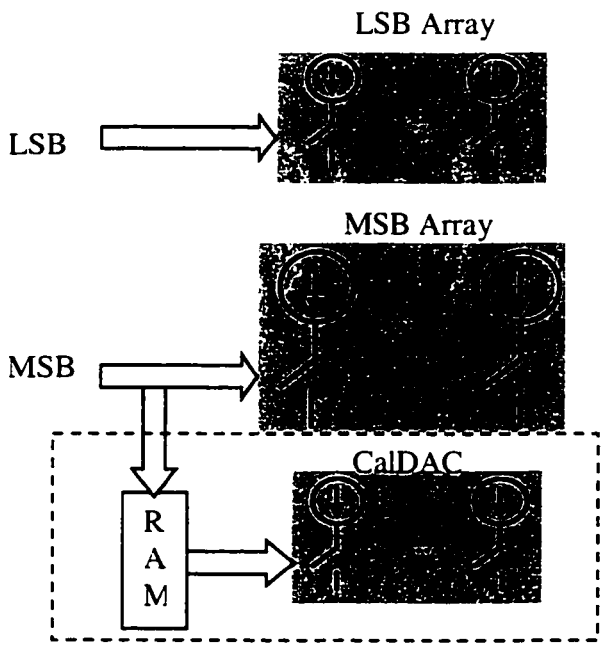
to correct the analog outputs. Since the errors are stored in static RAMs, not on floating gates, the calibration does not have to be done very often. In applications where DACs are not in heavy duty, foreground calibration can be used, in which the calibration is accomplished when the DACs are not in operation and during normal operation, the calibration circuitry will fall into sleeping mode. Compared to background calibrations. the power consumption for foreground calibration is negligibly small for a long run..

Digital calibration may be implemented in different ways. Some of them are analogous to the analog calibrations discussed above [55]-[58]. One such approach is shown in Figure 4.2 (a), where each current source to be calibrated has its own calibration DAC controlled by a register where the correction code is stored. Each calibration DAC can either adjust the gate voltages of the current sources or tune their currents. Like their analog counterparts, the calibration can be done in the background. However, the drawback of this structure is that the multiple calibration DACs are not "sufficiently" utilized while costing large area and power.

A more efficient approach is to employ a single calibration DAC controlled by a RAM where errors corresponding to each MSB code (assuming only the MSB array are calibrated), instead of each individual current source, are stored [59]-[62]. During the conversion time, the RAM are addressed by the digital inputs and read out to drive the calibration DAC which can be implemented by another current-steering DAC. An example of this approach is illustrated in Figure 4.2(b). The calibration scheme proposed in this chapter also belongs to this category. Compared to the calibration methods proposed in [59]-[62], the new scheme can calibrate both INL and DNL of the DAC, and meanwhile has the potential to compensate for nonliearities introduced by insufficient output impedance of the current sources. It is suitable for very low-voltage environment and easy to implement in a fully digital CMOS process leading to a considerable cost reduction.



(a)



(b)

Figure 4.2 Examples of Digital calibration
(a) each calibrated current source has its own CalDAC
(b) the overall DAC share a single CalDAC

4.2.2 A New Calibration Scheme

The basic idea behind this new calibration scheme is conceptually illustrated in Figure 4.3. The main DAC to be calibrated contains an nm-bit thermometer-decoded MSB array and an nl-bit LSB array. The latter may be binary-weighted or further segmented. Only the MSB array will be calibrated since the requirements for the LSB array are much relaxed and relatively easy to meet. The configuration in the calibration mode is shown in Figure 4.3(a). During the calibration, first the MSBs of the DAC are set to all "0" and the LSBs are set to all "1", consequently, the overall LSB array (including a dummy current source of value 1LSB) is switched to the positive output (V_{out+}) while the whole MSB array is switched to the complementary output (V_{out-}). The differential output (V_{out+}- V_{out-}) is then measured by a slow but highly accurate ADC and the result is saved and denoted as D_{LSB}. In the following calibration cycles, the LSB inputs are all set to "0". The MSB inputs are increased by 1 in each calibration cycle and meanwhile the accumulator is increased by D_{LSB}. For example, in the jth calibration cycle, the MSB input is equal to j, while the accumulator contains a value $j \cdot D_{LSB}$ correspondingly. The differential analog output is digitized by the ADC and the result is denoted as D(j). Ideally

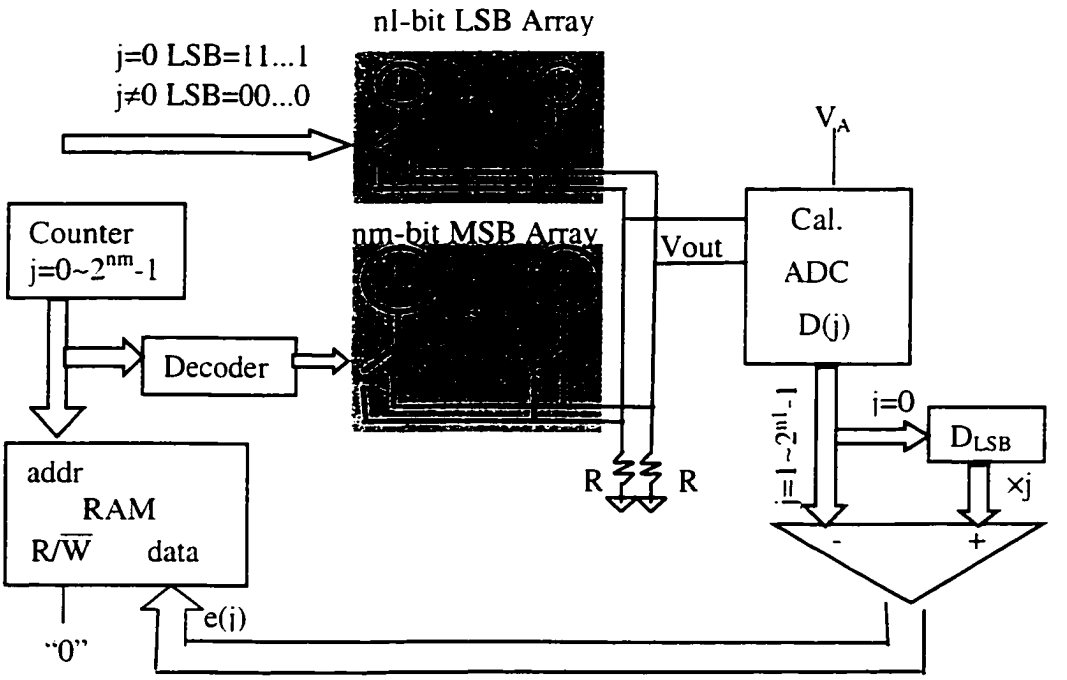
$$D(j)= j \cdot D_{LSB} \qquad (1 \le j \le 2^{nm}-1) \tag{4.1}$$

However, the error of D(j), denoted as e(j), is given by

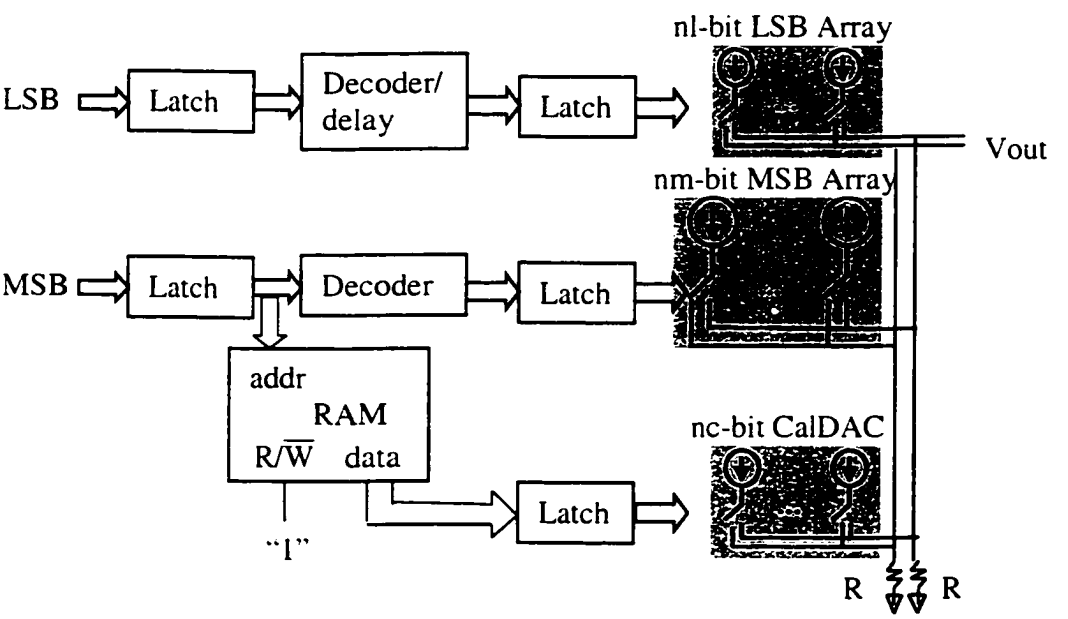
$$e(j)=D(j)- j \cdot D_{LSB} \qquad (1 \le j \le 2^{nm}-1) \tag{4.2}$$

It is stored as word j of the RAM. In this way, the error of each MSB code (from 1 to $2^{nm}-1$) is measured and stored in the corresponding location of the RAM.

In the conversion mode, as shown in Figure 4.3(b), the digital inputs drive the main DAC arrays and meanwhile the MSB inputs address the corresponding word of the RAM and



(a)



(b)

Figure 4.3 The conceptual block diagrams of the new calibration scheme
(a) calibration mode (b) conversion mode

read out the error code. The error code then drives the calibration DAC (CalDAC), which is also implemented by a current-steering array with differential outputs. The CalDAC generates a correction current that is summed with the current output of the main DAC to provide the overall output current.

In the above description, we ignored the gain mismatch between the ADC and the DAC. In practice, the input swing of the ADC should be slightly larger than the output swing of the DAC to avoid overflowing. For example, in the DAC, assuming the current of 1LSB is I, the differential output swing of the DAC is from -I·R·N to I·R·N and the common-mode voltage is equal to I·R·N/2. The ADC must be able to handle bipolar analog inputs in the range of $\pm V_A$, where $V_A > I \cdot R \cdot N$. When the digital input of the DAC is set to all "0", the differential output of the DAC is equal to -I·R·N. Since $-I \cdot R \cdot N > -V_A$, if offset binary is used for the output coding of the ADC, the digital output of the ADC will be larger than 0. It is actually the offset of the DAC with respect to the ADC, and denoted as D_{off}. During the calibration, this offset needs to be removed from all ADC digital outputs.

Since the ADC has a higher gain than the DAC, the error measured by the ADC, i.e. the previously mentioned $e(j)=j \cdot D_{LSB}-D(j)$, represents different analog values for the ADC and the CalDAC, therefore we can not use e(j) to directly drive the CalDAC. To compensate for the gain mismatch between the ADC and the DAC, as shown in Figure 4.4, for each MSB code (j) of the main DAC, a successive approximation process is undertaken to determine the correction code to be stored in the RAM and used to drive the CalDAC. This code is denoted as Dc(j). The bits of resolution assigned for each block in the figure will be derived in the next section. If the CalDAC has nc bits of resolution, the successive approximation will take nc steps, starting from the highest bit of the CalDAC. In each step of the successive approximation, the corresponding bit of the CalDAC is first set to "1" and then the differential output of the overall DAC including the main DAC and the CalDAC is digitized

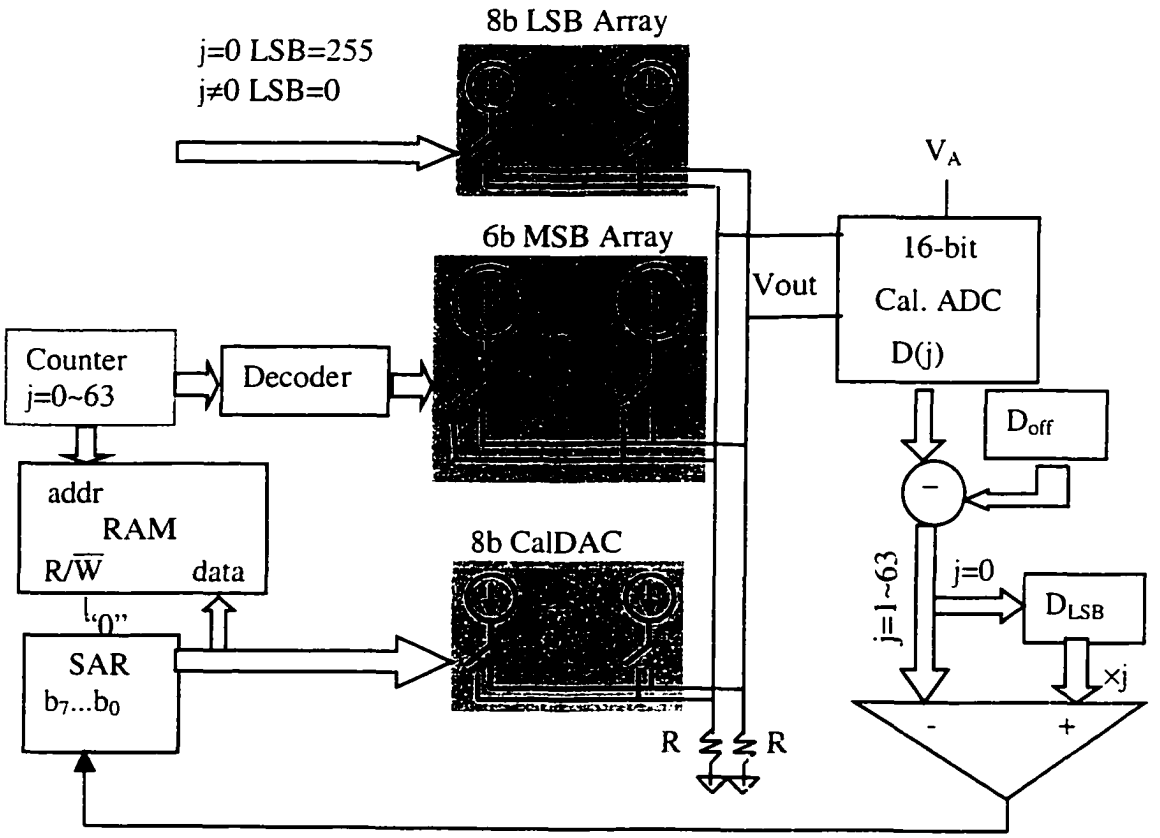


Figure 4.4 The Block diagram for calibrating the MSB array of the main DAC

by the ADC and compared to $j \cdot D_{LSB}$. If it is larger than $j \cdot D_{LSB}$, this bit is reset to "0", otherwise it remains to be "1". The same process is repeated for each bit of the CalDAC. The final result or the correction code Dc(j), is then stored in word j of the RAM and will eventually be read out during the conversion time to drive the CalDAC. Since the outputs of the CalDAC are differential, tuning can be done in both directions. When there is no error, in another word, when the output of the main DAC, D(j), happens to be equal to $j \cdot D_{LSB}$, the correction code, Dc(j), will be equal to 2^{nc-1} (=10...0 in binary), which means that half of the current sources in the CalDAC are switched to Vout+ while the other half are switched to

Vout-. To keep the two halves in balance, a dummy current source, whose current is equal to 1LSB of the CalDAC, is used and always connected to Vout-.

It can be observed that the MSB array of the main DAC operates in the same way in both calibration mode and conversion mode. Therefore, the current sources in the MSB array see similar drain voltages in both modes, except that during calibration, the LSBs of the DAC are all set to "0" while during conversion time, the LSBs may change from all "0" to all "1". However, the variations of LSB inputs only cause a minor change of the drain voltage of the current sources, which is no larger than $1/2^{nm}$ of the full-scale voltage. This implies that the new calibration method has the potential to compensate for errors due to insufficient output impedance of the current sources in the MSB array. As a result, it allows us to use simple and small current sources and less cascode stages, which is very attractive for low-voltage design. The output impedance requirement for the LSB array is much lower since it only conducts a $1/2^{nm}$ of the full-scale current. Meanwhile, small current also results in long-channel devices, and hence high output impedance.

Notice that this calibration algorithm is subject to "gain error accumulation". During the calibration, the total current of the LSB array, or D_{LSB} is accumulated and serves as the reference for the adjustment at each MSB code. If D_{LSB} has a small amount deviation, Δ, from its ideal value, i.e. $1/2^{nm}$ of the full scale where nm is the number of bits of the MSB array, during the calibration, this error will be accumulated as the MSB code increases. After the calibration, the full-scale current becomes $2^{nm}\cdot\Delta$ larger or smaller than it is before, resulting in a gain error. A more serious problem is that to compensate for this error, large tuning range is required for the CalDAC.

To limit the gain error accumulation and reduce the tuning range required for the CalDAC, a bias calibration as shown in Figure 4.5 is undertaken before calibrating the MSB array. Two bias voltages are generated by the bias generator, one is fixed to drive the MSB

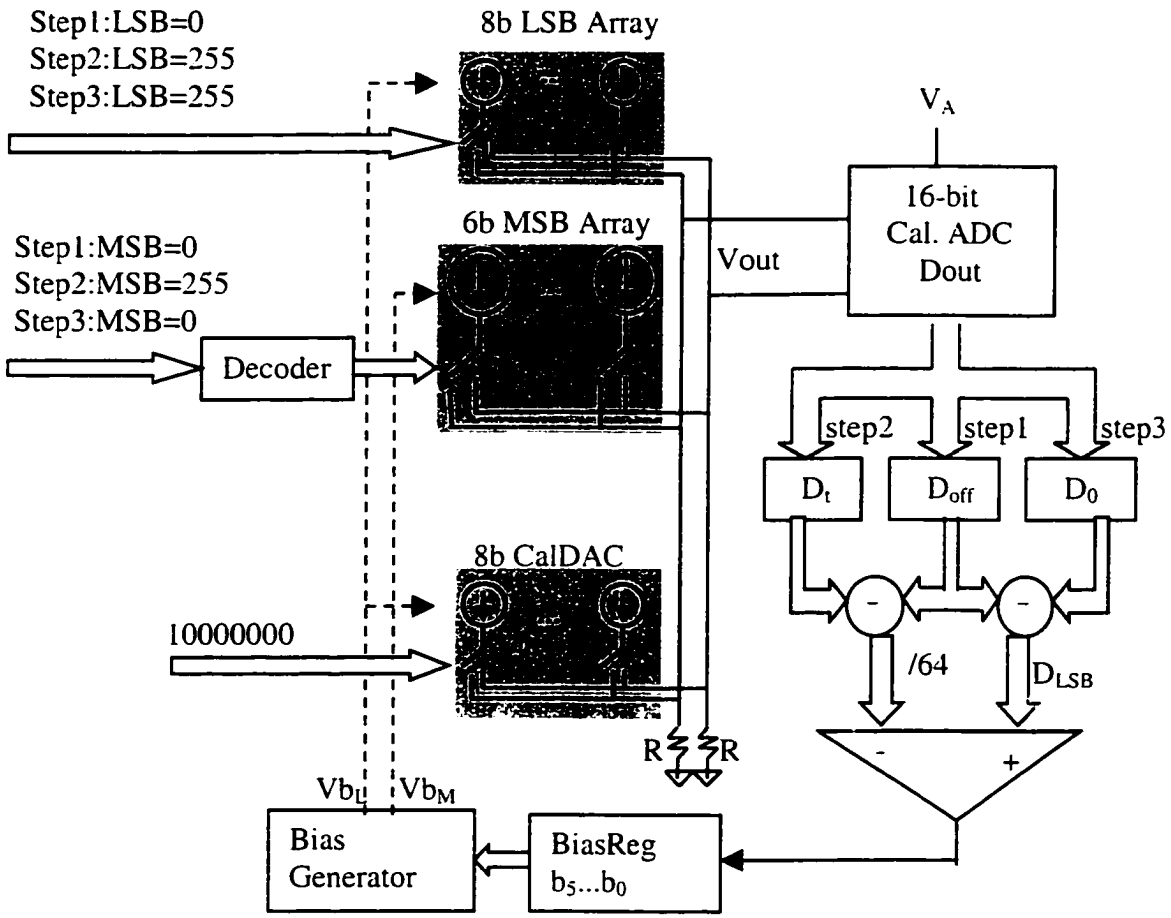


Figure 4.5 The block diagram for bias calibration

array and hence denoted Vb_M, while the other bias voltage Vb_L drives the LSB array and the CalDAC array. Vb_L is tunable through another current-steering DAC termed DAC_B, which is a part of the bias generator shown in Figure 4.5, and controlled by a register, BiasReg. The main goal of this calibration is to tune the bias voltage of the LSB array so that the total current of the LSB array, or D_{LSB}, is equal to $1/2^{nm}$ of the total current in the main DAC. During this calibration, the digital inputs of the nc-bit CalDAC are always set to 2^{nc-1}, i.e. the differential output of the CalDAC is approximately equal to 0. The bias tuning is accomplished by another successive approximation process starting with the MSB of

DAC_B and progressing one bit at a time to the LSB of DAC_B. Each approximation cycle contains three steps: First, set all bits of the main DAC to "0", the differential output, or the offset of the DAC, is digitized by the ADC and denoted as D_{off}; Second, set all bits of the main DAC to "1" so that the total current of the main DAC is digitized and denoted as D_t ; Third, set the MSB inputs to all "0" and the LSB inputs to all "1" (the dummy current source in the LSB array is also turned on), the differential output is digitized and denoted as D_0. Notice that $D_{LSB}=D_0-D_{off}$. Compare D_{LSB} to $(D_t- D_{off})/64$, if it is larger, the corresponding bit of the DAC_B remains set, otherwise, reset to "0". After the bias is calibrated, the calibration for each MSB code shown in Figure 4.4 is then started. The procedure of the overall calibration is summarized in Figure 4.6 and the block diagram in the conversion mode is shown in Figure 4.7.

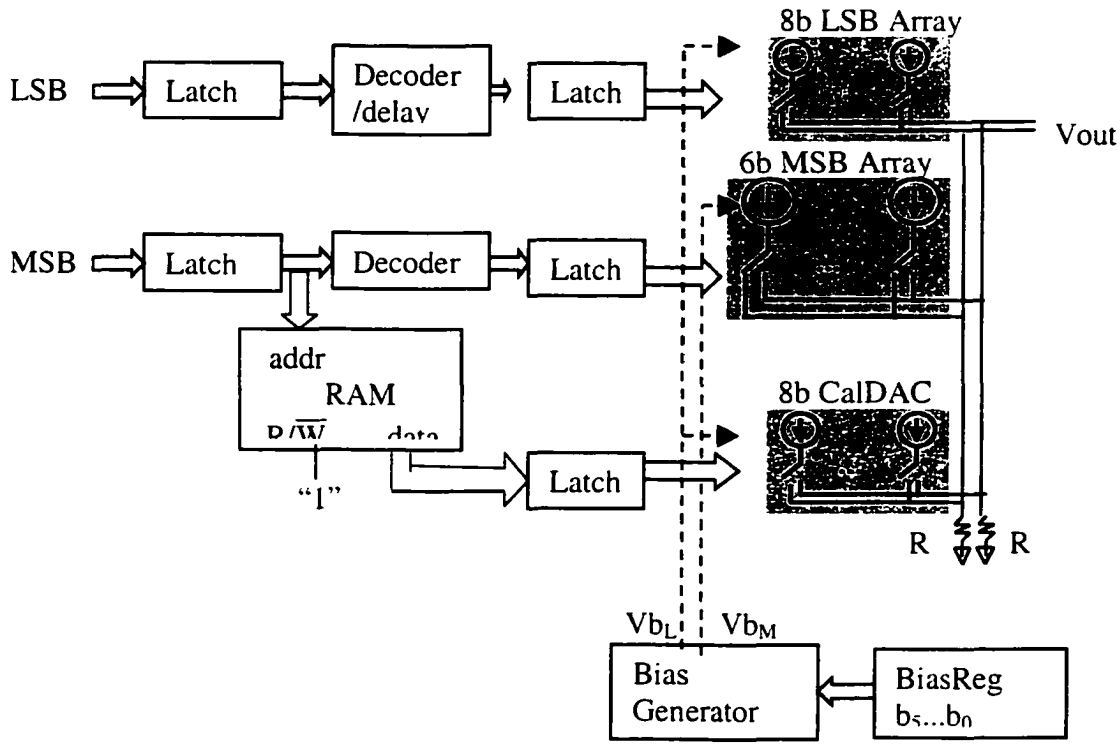


Figure 4.7 The conversion-mode block diagram

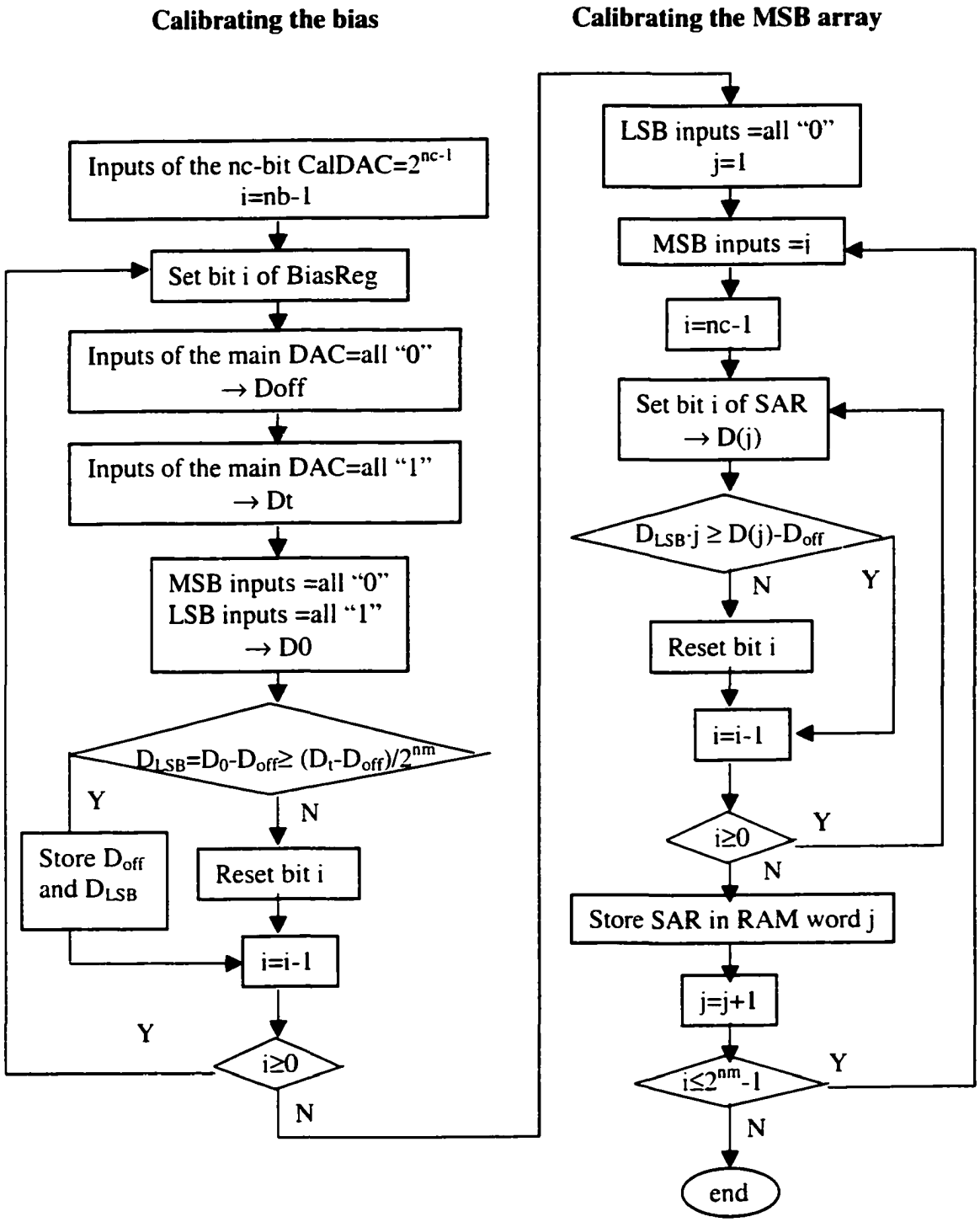


Figure 4.6 The calibration procedure

4.3 STATISTICAL ANALYSIS AND BEHAVIORAL MODELING

An important issue for current-steering DAC design is to optimize the segmentation and determine the matching accuracy, the output impedance and the size of the current sources of each DAC array to guarantee a given soft yield requirement. In this design, we also need to determine the resolution and accuracy of the ADC, the CalDAC and the bias tuner DAC_B. The DAC performance is strongly dependent on the optimization of these parameters. They will be estimated in the following three sections with the help of statistical analysis and Monte-Carlo simulations. A joint INL/DNL yield target of 99% for INL/DNL<0.5LSB will be established in the 0.13μm process.

In Section 4.3.1, design parameters are derived for an intrinsic 14-bit current-steering DAC using this process. It is shown that without calibration, current source array with very large gate area (1.4 mm^2) has to be used to overcome the random mismatch, thus achieve the 14-bit linearity and ensure high yield. Calibration scheme discussed in the previous section is analyzed in Section 4.3.2 and Section 4.3.3. In Section 4.3.2, only the MSB array is calibrated as shown Figure 4.4, where all the current arrays are sharing the same bias voltage. With this calibration, it is shown that the gate area of the current arrays can be reduced by a factor of 16. However, the analysis in Section 4.3.2 also shows that this calibration algorithm by calibrating the MSB array only suffers from gain error accumulation, which we have already mentioned in the previous section. This gain error accumulation effect becomes very significant when systematic errors are present, and hence limiting the reduction of the MSB array. To solve this problem, in Section 4.3.3, the bias calibration as shown in Figure 4.5 is used before calibrating the MSB array. It effectively limits the gain error accumulation so that the current sources in the MSB array may be further reduced and finally the total gate area of the current arrays are over 500 times smaller than that of the DAC without calibration.

Before we start the analysis, it is worth mentioning that in the following discussion, all currents are in the units of LSB and the unit LSB all refers to 1 LSB of the 14-bit DAC unless otherwise mentioned. The power supply to be used in the implementation is 1.5V and the full-scale current of this design is established to be 10mA to achieve a desired output swing of ±0.5V when driving 50Ω loads. Therefore, the current corresponding to 1 LSB is equal to $10mA/2^{14}=0.61uA$.

PMOS devices will be used to implement the current sources. One reason for using PMOS current sources is because they generate lower 1/f noise than their NMOS counterparts in the process available for this design. A more important reason of choosing PMOS devices is that in this process, they can be placed within an Nwell and hence partially isolated from the noisy digital circuitry that will be placed on the low-impedance substrate.

4.3.1 Without Calibration

We can start the design by considering a 14-bit current-steering DAC without any calibration. As we showed in Chapter 2, to achieve INL and DNL less than 0.5 LSB and guarantee 99% yield in the absence of any gradient effects, the relative standard deviation of each unit current source (σ_u) should be less than 0.22%. If the gate-source voltage (Vgs) of the current sources is -0.8V, to generate a current of 0.61uA and meet the matching requirement, the gate size of a unit current source ends up to be W_u/L_u=3um/28.8um. Therefore, the total gate area of the main DAC current arrays would be $(2^{14}-1)\cdot W_u \cdot L_u \approx 1.4mm^2$. This large gate area will introduce large layout parasitic capacitance which will seriously degrade the high-frequency performance of the DAC.

4.3.2 Calibration of the MSB Array

Based on the calibration algorithms described in the previous section, after calibration, the overall errors of the DAC will include both the residual errors of the calibrated MSB array and the intrinsic errors of the uncalibrated LSB array. Assuming that they are uncorrelated and conservatively assuming each of them contributes to a half of the overall error budget, the residual errors from the MSB and LSB array should be less than 0.25 LSB. Therefore, the nl-bit LSB array needs to maintain intrinsic nl+1-bit accuracy. The number of bits of the MSB array, nm, determines the size of the binary-to-thermometer decoders, latches and the RAM. For larger nm, more bits of the DAC will be calibrated, which may relax the design of the analog circuitry and result in smaller analog area, however the digital circuitry will increase exponentially with the increase of nm, which may finally becomes the dominant factor limiting the conversion rate and die size. Large digital circuitry also reduces the power efficiency of a DAC that is defined as the ratio of the power delived to the output to the total power dissipated.

A similar tradeoff as shown in [5] is made in this design based on the process characteristics. As a result, we segment the DAC as follows: the 6 MSBs drive a thermometer-decoded array called the MSB array, the 4 bits in the middle drive another thermometer decoded array called the Upper LSB or ULSB array, while the 4 lower LSBs drive a binary-weighted array termed the LLSB array. According to the calibration algorithm described in the previous section, we will calibrate only the 6-bit MSB array. Therefore, the 8-bit LSB array needs only to maintain 9-bit accuracy, which is relatively easy to achieve in today's technology and costs a reasonably small area. Based on (2.21), for an 8-bit DAC to achieve INL of 0.25 LSB and over 99% yield, the relative standard deviation of the unit current source (σ_u) can be 4 times larger than that of the uncalibrated 14-bit DAC discussed in Section 4.3.1, i.e. σ_u=0.88%. From equation (1.3), it is shown that the gate area ($W_u \cdot L_u$) of

the current sources can be reduced by a factor of 16. To maintain the same W/L ratio, W_u and L_u can both be reduced by a factor of 4, i.e. W_u=0.75μm, L_u=7.2μm.

As for the MSB array, since the errors of each MSB code are digitized by the ADC and the CalDAC, after calibration, the residual errors are mainly determined by the errors of the ADC, the CalDAC. For simplicity, we assumed the errors of the ADC and the CalDAC should both be limited within 0.25 LSB. Experimental results suggest that by doing so, we actually over-acted the error budget, since except these two factors, other mechanisms such as finite output impedance of the current sources, bias drifting etc. may also impact the effect of calibration. Considering that the analog input swing of the ADC is slightly larger than the full scale of the DAC, the ADC should have at least 16-bit resolution and accuracy to limit its error less than 0.25LSB. The CalDAC also needs to have 0.25LSB minimum resolution, but the INL requirement is less important. As long as it is monotonic and has no missing code, the successive approximation processes can proceed in the right direction. The full scale of the CalDAC depends on the maximum error possible for the MSB codes. In other words, it determines the maximum tuning range for the MSB array, and will be estimated in the following:

As we described in Section 4.2.2, the total current of the 8-bit LSB array (including a 1 LSB dummy current source) serves as the reference, I_{ref} for the MSB array. Ideally, $I_{ref}=2^8$ LSB. The 63 ($=2^6-1$) current sources in the 6-bit MSB array, denoted as Im_1, Im_2,..., Im_{63}, also each steers a current of 2^8LSB. If the unit current source in the 8-bit LSB arrays, as determined previously, has a gate width of W_u=0.75μ and gate length L_u=7.2μm to maintain 9-bit accuracy, and if the same gate length is used, i.e. $L_m=L_u$=7.2μm, with a current ratio of 2^8, the gate width of the unit current source in the MSB array is $W_m=2^8 \cdot W_u$=192μm. Assuming only random error are present, I_{ref} has a relative standard deviation of $\sigma_u / \sqrt{2^8}$,

where σ_u=0.88%, and so does each unit current source in the MSB array. Therefore the jth current source in the MSB array can be expressed as

$$I_{M,j} = 2^8(1+\delta_{M,j}) \qquad (1\le j\le 2^6-1) \tag{4.3}$$

where $\delta_{M,j}$~N(0, $\sigma_u/\sqrt{2^8}$). During the calibration, if MSB=k, $1\le k\le 2^6-1$, the output current I(k), which is the summation of $I_{M,1}$, $I_{M,2}$,..., $I_{M,k}$, will be compared to k· I_{ref}. Ignoring the errors of the ADC and CalDAC, the error for MSB=k, or e(k) can be expressed as

$$e(k) = k\cdot I_{ref} - \sum_{j=1}^{k} I_{M,j} \tag{4.4}$$

Therefore, the standard deviation of e(k) is

$$\sigma_e(k) = \sqrt{k^2\cdot 2^8 + k\cdot 2^8}\cdot\sigma_u \tag{4.5}$$

which reaches its maximum value when $k=2^6-1$. To guarantee in over 99% of the chances, the errors of each MSB code are still in the tuning range of the CalDAC, the full swing of the CalDAC is chosen to be 3·max(σ_m(k)), that is $3\sqrt{(2^6-1)^2\cdot 2^8+(2^6-1)\cdot 2^8}\cdot\sigma_u \approx 27\text{LSB}$. Recall that the minimum resolution of the CalDAC is 0.25LSB, therefore it must have at least 5+2 bits of resolution.

Notice that in (4.4), the first term under the square root is much larger than the second term when $k=2^6-1$. This is because in e(k) given by (4.4), I_{ref} are accumulated k times, while the k current sources of the MSB array are uncorrelated from each other so that their deviations are averaged out when being summed together. This result again shows the significant "gain error accumulation" effect inherent in this calibration algorithm. (4.5) implies that we can further reduce the current sources in the MSB array. By doing so, the standard deviations of the current sources increases, so does the second term of (4.5). As long

as the first term of (4.5) is still dominant, $\sigma_e(k)$ and hence the tuning range of the CalDAC will not increase much.

However, a serious problem for further reduction of the current sources in the MSB array is that by doing so, the channel length of the current sources in the MSB array will be less than that of the current sources in the LSB array, therefore causing systematic mismatches between the LSB array and the MSB array. As previously mentioned, due to the significant "gain error accumulation" effect inherent in this calibration algorithm, the systematic mismatch between the total current in the LSB array (I_{ref}) and the current sources in the MSB array will be accumulated during the calibration, yielding large errors as the MSB code increases. To compensate for these accumulated gain error, an unreasonably large CalDAC is required.

4.3.3 Calibration of the Bias Voltage

To overcome the gain error accumulation effect and further reduce the current source arrays, bias calibration is used. As described in Section 4.2.2, the goal of the bias calibration is to tune the bias voltage of the LSB array through DAC_B so that the total current of the LSB array, or I_{ref}, is approximately equal to 1/64 of the total current of the main DAC. In other words, after this calibration, I_{ref} is approximately equal to the average current of the unit current sources in the MSB array. This calibration limits the possible gain error that may be introduced when calibrating the MSB array and reduces the errors of each MSB code (see e(k) in (4.4)). Therefore, a small tuning range is required for the CalDAC. From another point of view, using the same CalDAC, the bias calibration allows us to further reduce the size of current sources in the MSB array.

From Section 4.3.2, we know that the ideal value of I_{ref} and each unit current sources in the MSB are equal to 2^8 LSB. With gate length L_u=7.2μm and gate width

$W_m=2^8 \cdot W_u=192\mu m$, the relative standard deviation of the current source is $\sigma_u/\sqrt{2^8}$, where $\sigma_u=0.88\%$. Assuming the gate width and length of the current sources in the MSB array are reduced simultaneously by a factor of 2^p, based on the expression shown in (1.4), the relative standard deviation of the current sources are increased by a factor of 2^p, i.e. in (4.3),

$$\delta_{M,j} \sim N(0, \sigma_u/\sqrt{2^{8-2p}}) \tag{4.6}$$

Since the W/L ratio of the current source does not change, the current they are conducting will not change either.

After the bias calibration, ignoring the residual errors due to the DAC_B and the ADC, the total current of the LSB array, or I_{ref}, is approximately equal to the average current of the unit current sources in the MSB array, which can be expressed as

$$I_{ref} \approx \bar{I}_M = \frac{1}{2^6-1}\sum_{j=1}^{2^6-1} I_{M,j} \tag{4.7}$$

From (4.3), (4.4) and (4.6), when calibrating the MSB array, the error for MSB=k, $1 \le k \le 2^6-1$ is equal to

$$e(k) \approx 2^8 \cdot \left(\frac{k}{2^6-1}\sum_{j=k+1}^{2^6-1}\delta_{M,j} - \frac{2^6-1-k}{2^6-1}\sum_{j=1}^{k}\delta_{M,j}\right) \tag{4.8}$$

where $\delta_{M,j} \sim N(0, \sigma_u/\sqrt{2^{8-2p}})$. The standard deviation of e(k) is equal to

$$\sigma_e(k) \approx 2^{p+1}\sqrt{k(2^6-1-k)} \cdot \sigma_u \tag{4.9}$$

As we expected, the maximum value of (4.9) occurs at the middle scale when $k=2^6/2$ and is approximately equal to $2^{6+p}\sigma_u$. The full scale of the CalDAC is chosen to be $3\times 2^{6+p}\sigma_u$ LSB. In this design, we choose p=3. Since $\sigma_u=0.88\%$, the full scale of the CalDAC is 3×4.5 LSB, hence it needs at least 4+2 bits of resolution.

With p=3, the gate width and length of the MSB current sources are further reduced by a factor of 8, i.e. L_m=0.9μm and W_m=24μm. As a result, the channel length of the current sources in the MSB and LSB array are different. Simulation shows that the systematic mismatch between a single current source in the MSB array and the total current of the LSB array (including a 1 LSB dummy current source), or I_{ref}, is around 1.6LSB. As it will be shown, this systematic mismatch can be effectively compensated through the bias calibration.

The size of the current sources used in the MSB and LSB arrays are summarized in Table 4.2. It is shown that through the above calibration, the total gate area of the main DAC array is only 2743.2μm^2, about 510 times smaller than it is without any trimming and calibration.

Table 4.2 Comparison of current sources used in a 14-bit DAC with and without calibration

		6b MSB array	4b ULSB array	4b LLSB array	Gate area (mm^2)
Decoding		Thermometer -decoded	Thermometer-decoded	Binary-weighted	
Number of current sources		63	15	4 (b3 b2 b1 b0)	
Current sources W(μm)/L(μm)	Without calibration	768/28.8	48/28.8	24/28.8 12/28.8 6/28.8 3/28.8	1.4
	Calibrate the MSB array only	192/7.2	12/7.2	6/7.2 3/7.2 1.5/7.2 0.75/7.2	0.0885
	Calibrate the bias before calibrate the MSB array	24/0.9	12/7.2	6/7.2 3/7.2 1.5/7.2 0.75/7.2	0.0027

The bias generator used is shown in Figure 4.8, where the full-scale current of the DAC is set by an external voltage Vext and an external resistor Rext. The external Opamp set up the bias voltage for three NMOS transistors (Mn_X, Mn_M and Mn_L) so that the current

going through Mn_X is equal to Vext/Rext. This current is then mirrored through Mn_M and Mn_L to sink the cascode diode connected PMOS transistors, yielding two bias voltage Vb_M and Vb_L. Vb_M is a fixed bias voltage driving the MSB current array, while the bias voltage Vb_L is tunable and driving the LSB array and the CalDAC array. Tuning Vb_L is accomplished through another current-steering DAC, DAC_B. The cascode diode connected PMOS transistors in the DAC_B, controlled by a register BiasReg, are either switched to the

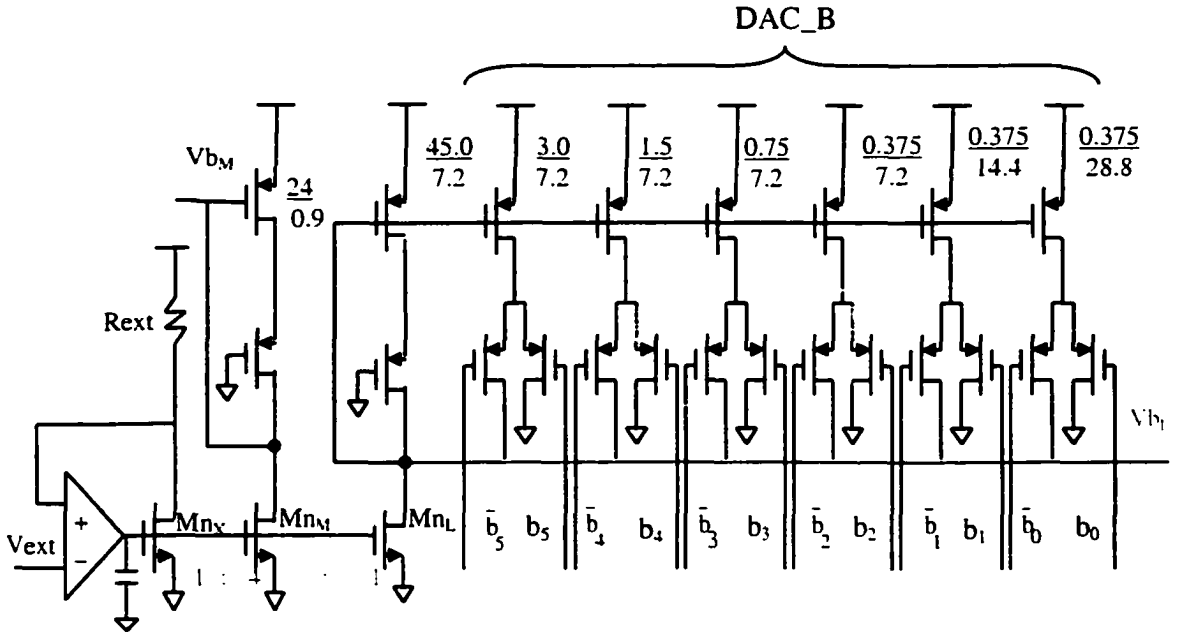


Figure 4.8 Bias Generator

ground or to the NMOS current source Mn_L. When they are switched to the ground, Vb_L drops and when they are connected to the NMOS current source Mn_L, Vb_L increases. These switchable PMOS transistors are binary weighted. The cascode PMOS transistor used to generate the bias voltage Vb_L match the current sources in the LSB array and form an array denoted as the LSB bias array. The DAC_B is a small portion of this array. Correspondingly,

the PMOS transistors used to generate the bias voltage Vb_M match the current sources in the MSB array and hence the array they form is termed MSB bias array.

Assume the MSB bias array consists of k_1 diode connected MSB unit current sources and in the LSB bias array, the cascode diode connected PMOS transistors that are normally connected to the NMOS current source Mn_L are equivalent to k_2 ULSB unit current sources. Based on Table 4.2, in Figure 4.8, k_1=1 and k_2=4. The details on how to determine k_1 and k_2 will be given later. Ignoring the mismatch between the NMOS transistors Mn_M and Mn_L, the total current drawn from the MSB bias array is equal to $k_1 \cdot 2^8$ LSB, while the total current draw from the LSB bias array is equal to $k_2 \cdot 2^4$ LSB. In the LSB bias array, PMOS transistors that are always connected to the NMOS current source Mn_L are equivalent to 3.75 ULSB unit current sources. The other switchable cascode PMOS transistors in the DAC_B are totally equivalent to 0.5 ULSB unit current source. Before the bias calibration, the initial inputs of the DAC_B are set to "100000", thus, the transistor equivalent to 0.25 ULSB unit current source is connected to Mn_L, while the others are be connected to the ground. Therefore, normally the cascode PMOS transistors connected Mn_L are equivalent to 4 ULSB unit current source, however, during the calibration, his number can be adjusted between 3.75 to 4.25, and consequently, the bias voltage Vb_L can be adjusted in both directions.

The determination of k_1 and k_2 as well as the tuning range of the DAC_B array depends on the matching between the bias arrays and the DAC arrays. First consider the MSB bias array. According to Table 4.2, a MSB unit current source ideally conducts a current of 2^8 LSB. When only random variations are present, with an ideal bias voltage, the current conducted by the jth equivalent MSB unit current source in the MSB bias array can be expressed as

$$I_{bM.j} = 2^8(1+\delta_{bM.j}) \qquad (1 \le j \le k_1) \qquad (4.10)$$

where the relative standard deviation $\delta_{bM,j} \sim N(0,\sigma_u/2)$ (see (4.6) where p=3). Likewise, with an ideal bias, the jth current source in the MSB array can be expressed as

$$I_{M,j} = 2^8(1+\delta_{M,j}) \qquad (1 \le j \le 2^6-1) \tag{4.11}$$

where $\delta_{M,j} \sim N(0,\sigma_u/2)$. Ignoring the mismatch between the NMOS transistors Mn_L and Mn_M, the total current of the MSB bias array is forced to be $k_1 \cdot 2^8$, that is

$$\sum_{j=1}^{k_1} I_{bM,j} = k_1 \cdot 2^8 \tag{4.12}$$

This current is mirrored through the MSB bias array to the MSB array of the main DAC. The actual current obtained in the MSB array does not exactly match the original input of the current mirror. This mismatch is related to the random variation of the current sources in both arrays. Therefore, through this current mirror, the actual current of the jth current source in the MSB array can be expressed as

$$I_{M,j} = \frac{k_1 \cdot 2^8}{\sum_{i=1}^{k_1} 2^8(1+\delta_{bM,i})} \cdot 2^8(1+\delta_{M,j}) \tag{4.13}$$

Similarly, a ULSB current source ideally conducts a current of 2^4LSB. When only random variation is present, with an ideal bias, the current conducted by the jth equivalent ULSB unit current source in the LSB bias array can be expressed as

$$I_{bL,j} = 2^4(1+\delta_{bL,j}) \qquad (1 \le j \le k_2) \tag{4.14}$$

where $\delta_{bL,j} \sim N(0, \sigma_u/\sqrt{2^4})$. Initially, k_2 such current sources are connected to the NMOS transistor Mn_L. During the bias calibration, a small amount of them with equivalent current equal to I_x are switched in or out of the bias array to adjust the bias voltage Vb_L. Since the adjustment can be done in both directions, I_x can be either positive or negative. After

calibration, the current drawn from the remaining PMOS transistors that are connected to Mn_L are equal to $k_2 \cdot 2^4$ LSB, therefore we have

$$\sum_{j=1}^{k_2} I_{bL,j} + I_x = k_2 \cdot 2^4 \tag{4.15}$$

where the mismatch between Mn_L and Mn_M is ignored.

On the other hand, with an ideal bias, the total current of the LSB array, or I_{ref}, is approximately equal to the summation of 2^4 ULSB unit current sources and can be expressed as

$$I_{ref} = \sum_{j=1}^{2^4} 2^4 (1 + \delta_{L,j}) \tag{4.16}$$

where $\delta_{L,j} \sim N(0, \sigma_u / \sqrt{2^4})$. When driven by the LSB bias array, the current of Mn_L is mirrored through the LSB bias array into the LSB array in the main DAC. In this case, the actual total current of the LSB array, or I_{ref}, can be expressed as

$$I_{ref} = \frac{k_2 \cdot 2^4}{\sum_{j=1}^{k_2} 2^4 (1 + \delta_{bL,j}) + I_x} \cdot \sum_{j=1}^{2^4} 2^4 (1 + \delta_{L,j}) \tag{4.17}$$

Recall that the goal of the bias calibration is to have I_{ref} close to $1/2^6$ of the total current of the main DAC, which is approximately equal to the average current of the 2^6-1 current sources in the MSB array. From (4.7), (4.13) and (4.17), the current being switched during the bias calibration, I_x, can be estimated as

$$I_x = \left[\frac{1}{2^4} \sum_{j=1}^{2^4} \delta_{L,j} + \frac{1}{k_1} \sum_{j=1}^{k_1} \delta_{bM,j} - \frac{1}{2^6 - 1} \sum_{j=1}^{2^6 - 1} \delta_{M,j} - \frac{1}{k_2} \sum_{j=1}^{k_2} \delta_{bM,j} \right] \cdot k_2 \cdot 2^4 \tag{4.18}$$

Therefore, the standard deviation of I_x can be expressed as

$$\sigma_x = \sqrt{k_2^2(2+\frac{2^6}{k_1}) + k_2 2^4} \cdot \sigma_u \tag{4.19}$$

The tuning range of the DAC_B is chosen to be $\pm 3\max(\sigma_x)$. It can be shown that the tuning range of the CalDAC or the maximum errors met when calibrating the MSB array has little dependence on the bias array.

In the above estimation, we ignored the quantization errors of the ADC and the DAC_B. However, to determine the size of the CalDAC, these factors have to be considered together with the random errors. Since the minimum resolution of the ADC is 0.25LSB, after the bias calibration, I_{ref} may have a deviation from $\bar{I}_M$ as large as 0.25LSB. According to (4.4), this systematic error will accumulate when calibrating the MSB array and reach its maximum value, $0.25\times(2^6\text{-}1)$ LSB when MSB=2^6-1. To compensate for this systematic error, at least a 4+2-bit CalDAC is needed.

Likewise, assuming the DAC_B has a minimum resolution of I_Δ, when this error is mirrored to the main DAC, it may result in an error in I_{ref} as large as $I_\Delta \cdot 2^8/(k_2 \cdot 2^4)$. The error of I_{ref} will be accumulated when calibrating the MSB array, and reaches the maximum value of $(2^6-1)\cdot I_\Delta \cdot 2^8/(k_2 \cdot 2^4) \approx 2^{10} \cdot I_\Delta/k_2$ when MSB=2^6-1. To reduce this error, large k_2 and small I_Δ are preferred. This not only helps to reduce the area and the number of bits of the CalDAC, but also reduces the number of successive approximation cycles needed for calibrating the MSB array. On the other hand, large k_2 and small I_Δ will increase the size of the LSB bias array, and as shown in (4.19), it will also increase the number of bits needed for the DAC_B.

The above estimations provide a good guideline for optimizing the design parameters. However, a final decision has to be made based on Monte-Carlo simulations and careful tradeoffs between area, power consumption and many practical issues. Besides, enough margins have to be left so that the design can tolerate any possible process variations. A

behavioral model was built using Matlab to emulate the overall calibration and conversion operations in the presence of random current mismatches and finite output impedance. The three parameters k_1, k_2 and I_Δ are finally chosen as follows: k_1=1, k_2=4 and I_Δ=0.125LSB. The complete bias circuitry is shown in Figure 4.7. As a result, the DAC_B has a full tuning range of ±4LSB and a minimum resolution of 0.125LSB, hence it has 6 bits of resolution, while the CalDAC array has 6+2 bits of resolution. The NMOS transistors Mn_M and Mn_L have a 4:1 ratio and are designed with mismatches negligibly small compared to those of the PMOS arrays. Simulation results by using the behavioral mode are shown in Figure 4.9 indicating that the yield for INL and DNL less than 0.5LSB is over 99%.

In summary, the overall calibration is accomplished by a bias calibration followed by a calibration of the MSB array. The block diagrams for the two calibration modes are given

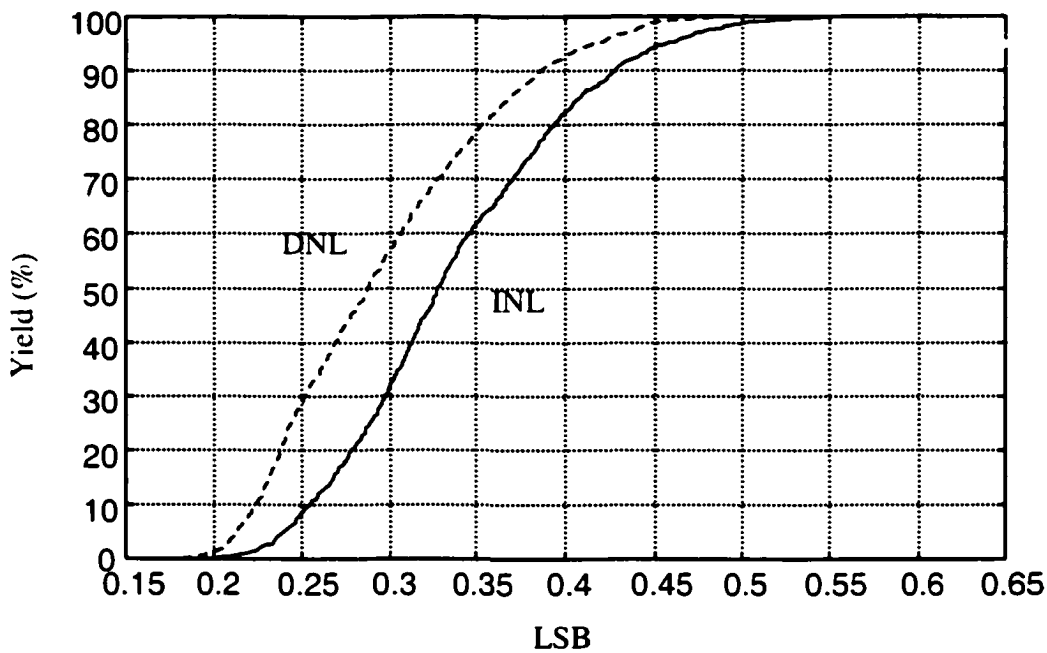


Figure 4.9 Simulated Yield of INL and DNL

in Figure 4.5 and Figure 4.4 respectively, and the overall calibration procedure is shown in Figure 4.6. The size of current sources and the bias generator can be found in Table 4.2 and Figure 4.8 respectively. In the next section, the circuit implementation of some critical building blocks will be described.

4.4 PROTOTYPE IMPLEMENTATION

4.4.1 Current Cells

As shown in Figure 4.10, each current cell contains a current source and a pair of switches controlled by a decoding logic (when thermometer decoded) or a delay cell (when binary weighted) followed by a latch & switch driver. The switches when turned on are operating in the saturation region, serving as a cascode stage to enhance the output impedance. For different current cells, the switches are scaled accordingly to keep the drain voltage of the current sources equal. Limited by the power supply voltage, more cascode stages are not affordable in this design.

The latch & switch driver shown in Figure 4.10 provides two functions: first, it synchronizes the digital signals before they feed into the current cells in order to avoid the timing skews due to the various delays between the outputs of the decoders, the delay cells and the RAM; second, it minimizes the fluctuation at the common node X of the switches during the transition. This node usually has a large parasitic capacitance due to the large dimensional current source and switches. Limiting its voltage variation reduces the charge and discharge going on this node, hence speeds the settling and reduces the glitches at the outputs.

In this work, the above functions are realized by a compact latch shown in Figure 4.11 [13]. The falling transition in this latch always starts before the rising transition. The

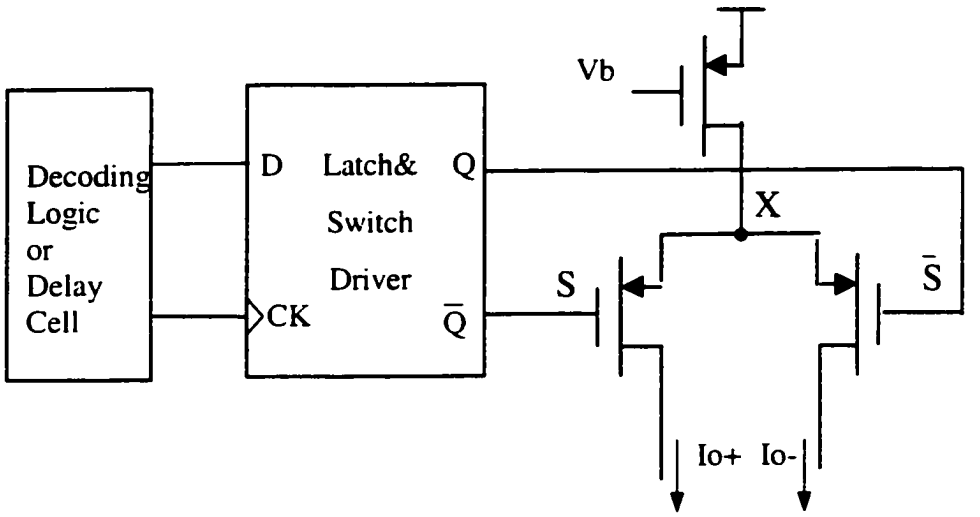


Figure 4.10 Current Cell

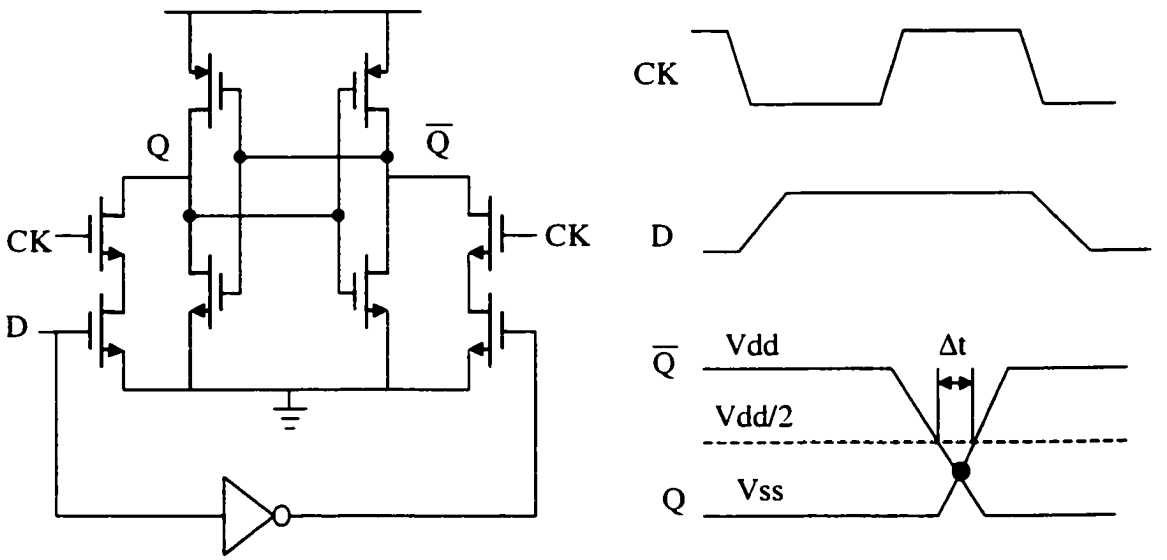


Figure 4.11 Latch & Switch Driver

The intrinsic slight delay between them lowers the cross points of the two switching control signals, thus avoids turning the two switches off simultaneously and reduces the fluctuation at the common node during the transition. To drive different load capacitance, the latches in the current cells are carefully ratioed and sized to keep the switching time equal and short.

4.4.2 RAM and Calibration Circuitry

The 64 word×8 bit static RAM is designed using a conventional method described in [63]. The RAM is written only when calibrating the MSB array of the DAC, thus the writing operations can be very slow, while the reading operations occur in the conversion mode and are in the same speed as the sampling frequency. The calibration control logic circuitry are synthesized using Verilog, Synopsis, and Siliconensemble in Cadence.

Since the main goal of this prototype is to verify the DAC calibration concept, for simplicity, the calibration ADC is implemented off-chip using a commercial product of Analog Device—AD7715 (16bit Σ-Δ ADC) [64]. Actually, the calibration ADC in this design converts only DC signals. It can be simply implemented using a low-order 16-bit Σ-Δ modulator with very high oversampling ratio. Due to the relatively low requirements on the analog circuitry, quite a few low-voltage high-resolution Σ-Δ modulators and ADC's have been reported in the literature [65]-[71]. Since a large portion of a Σ-Δ modulator are digital circuits, benefiting from the shrinkage of process, it costs small area and power consumption.

4.4.3 Layout

The performance of a current-steering DAC strongly depends on the layout of the current cell arrays. The floor plan of the current cell arrays including the bias generator in this design is illustrated in Figure 4.12, where all the current sources are placed at the center forming a big array, and surrounding this current source array are the switches, latches and

decoders. Within the current source array, the standard LSB current sources are placed in the middle, while the standard MSB current sources are partitioned into two and located on the top and bottom rows respectively. The switches/cascode stages, latches and decoders associated with the LSB arrays are placed to the left and right sides of the current source array, and connected to the corresponding current sources using horizontal lines. Likewise, the switches and the digital circuits associated with the MSB arrays are placed to the top and bottom of the current source array, hence the interconnections between them and the MSB current sources are in vertical.

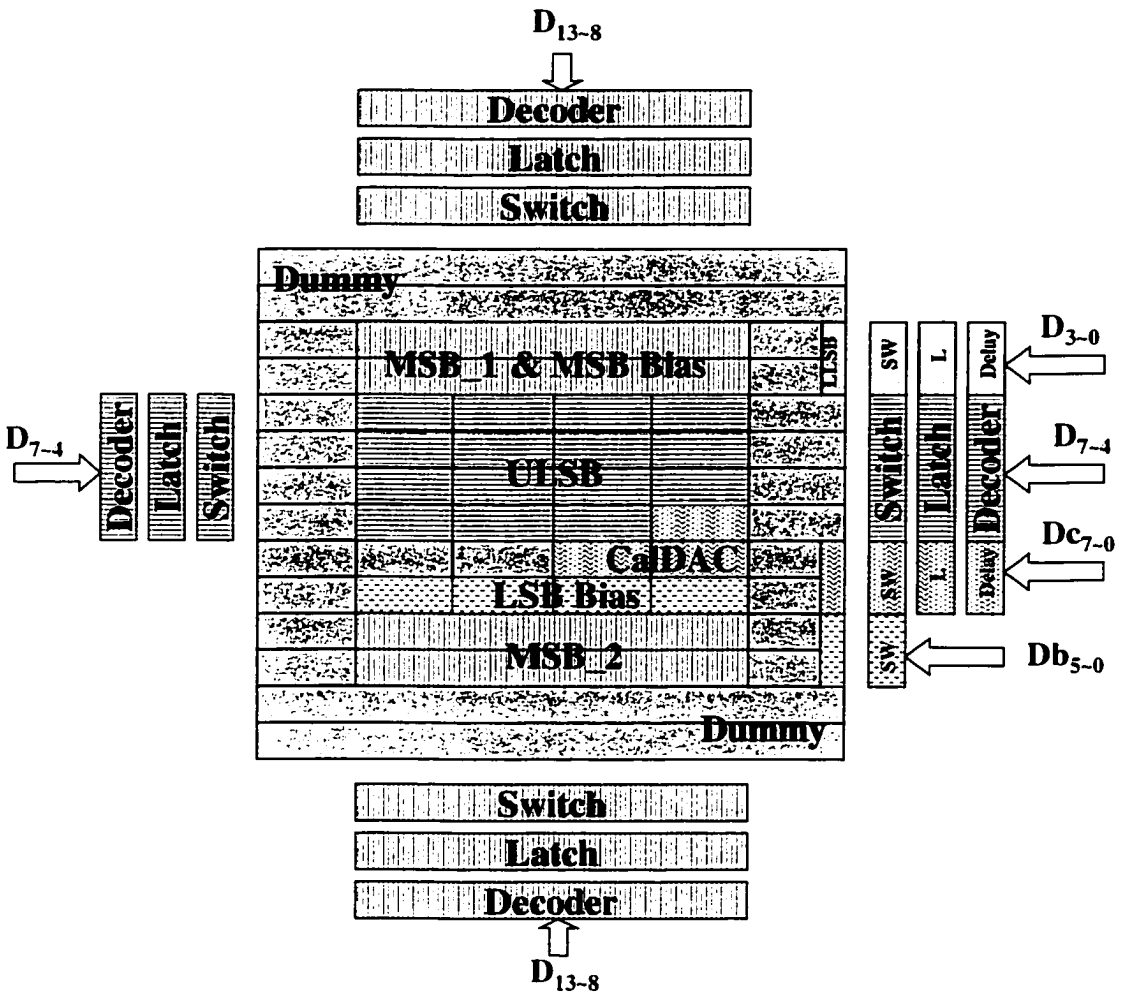


Figure 4.11 Floor plan of the main DAC array

The main goal to choose this arrangement is to minimize the interconnections between the current sources and their switches. As we mentioned in Section 4.1, minimizing the parasitic capacitance at the interconnecting nodes is essential for improving the settling and dynamic linearity of the DAC. Second, since the LSB array of the DAC, especially the 3-bit ULSB array, has to maintain 9-bit linearity without calibration, the current sources in the array are placed as close as possible in order to minimize the gradient effects. To further reduce the gradient effects, the switching sequences shown in Figure 4.13(a) are used.

Third, the current sources in the MSB array are to be calibrated, so are their gradient errors. To minimize their interconnections to the switches, the 64 current sources including

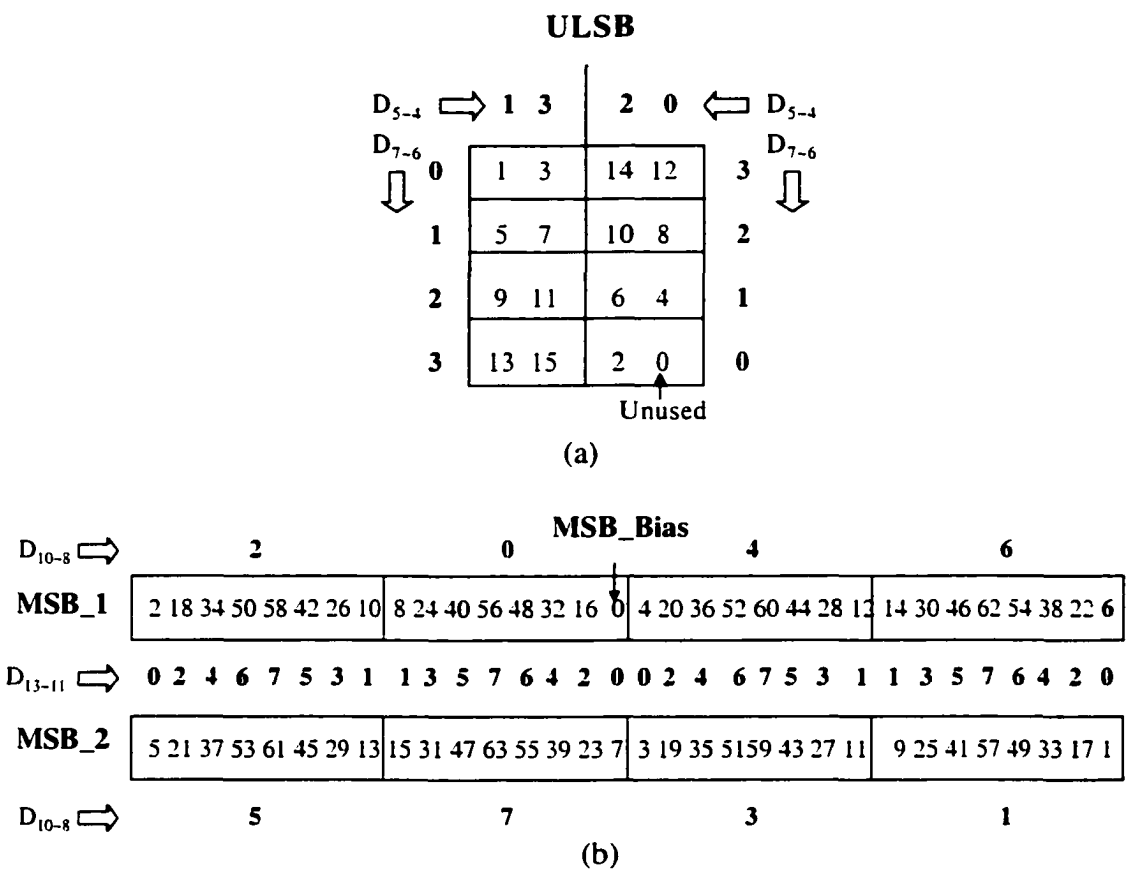
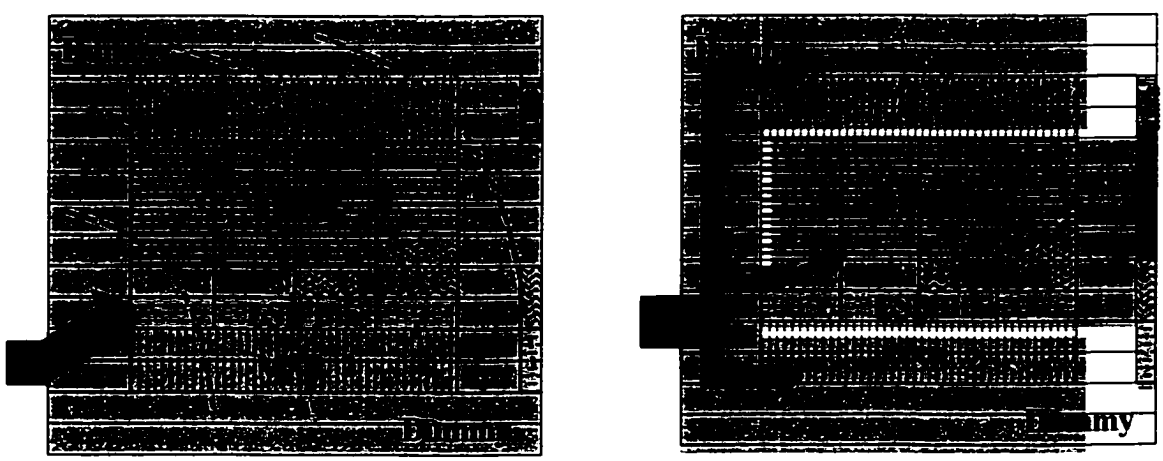


Figure 4.12 Switching sequence used for (a) ULSB array and (b)MSB array

that in the MSB bias array are divided into two rows on the top and bottom of the current source array respectively, so that they can be directly connected the switches using short vertical lines. Without much expense, the switching sequence shown in Figure 4.13(b) will help to reduce the gradient errors.

Minimizing the voltage drop along the power lines is also very critical for limiting gradient effects. To avoid crossing the digital circuits surrounding the current source array, the analog Vdd, enters from the left lower corner of the current source array, and covers theoverall current source array with multiple layers of metals. However, for a power distribution as shown in Figure 4.14(a), when the full-scale current is as high as 10mA, the voltage drop along the LSB array is still too high to achieve 9-bit accuracy. This is because the current drawn by the MSB current sources on the top row, which is approximately equal to a half of the full-scale current, also goes through the LSB array. Since that the MSB array of the main DAC are to be calibrated while the LSB array are not, it is essential to minimize



(a) (b)

Figure 4.14 Analog Vdd distributions (a) covers the overall current source array (b) separates the power lines of the MSB and LSB arrays

the voltage drop along the power line in the LSB array. Considering the LSB array only draws about 1/64 of the total current from the analog Vdd, the power line distribution shown in Figure 4.14(b) separates the power lines of the MSB and LSB arrays, therefore dramatically reduces the voltage drop in the LSB array.

In addition, to immunize common mode noise, the differential outputs are routed next to each other as close as possible. Since they are surrounding the whole current source array, and conducting large current, wide metal are used in order to reduce the parasitic resistance and hence the voltage drop to ensure all the transistors in the current cells are always in saturation region.

Other well-known layout techniques such as placing dummy current sources around the current source arrays to improve matching, using clock trees to reduce timing skew, separating the analog power supplies from the digital supplies to minimize noise coupling, etc are also employed.

4.5 EXPERIMENTAL RESULTS

A microphotograph of the chip is shown in Figure 4.15, where the 14-bit DAC is sharing the 3mm*3mm die with some other irrelevant designs. The active area of the DAC as indicated in the figure is only $0.1mm^2$ in a 0.13μm, 1.5V digital CMOS process. The current cell arrays illustrated in Figure 4.12 are on the left side of the DAC, the SRAM and the calibration control logic are on the right side, and in the middle are the clock drivers and the input registers. It can be seen that the area occupied by the current source arrays is comparable to the digital portions of the DAC.

Figure 4.16 and Figure 4.17 show the typical INL and DNL plots measured before and after calibration. As expected from the statistical analysis and simulations, before calibration, the DAC has 9~10 bits of linearity and the major errors are coming from the 6-bit

MSBs. After calibration, both INL and DNL are below 0.5LSB, therefore 14-bit linearity is achieved.

Figure 4.18 and Figure 4.19 show the typical post-calibration spectrums of full-scale differential outputs with doubly terminated 50Ω load and 100MS/s sampling frequency. The input sinusoids in the two figures are at a frequency of 0.9MHz and 41.5MHz respectively. It is shown that after calibration, the SFDR at lower signal band is over 82.5dB, and drops down to 62.5dB when the signal frequency is close to the Nyquist rate. The spectrum of two tones at 23.5MHz and 24.4MHz is shown in Figure 4.20, where the SFDR is 66dB.

As a summary, the typical SFDR of the differential outputs at sampling rate of 50MHz, 100MHz are plotted in Figure 4.21, where the dashed lines are the SFDR before the calibration and the solid lines are those after the calibration. Since the SFDR at low-frequency band is mainly determined by the static current mismatch, before calibration, the SFDR is only around 63dB, while after calibration it is boosted up to 82dB. For high-frequency signals, the improvement of SFDR with calibration becomes much less. This is because the dynamic nonlinearity dominates the static errors and therefore the benefit of calibration becomes less significant. Even so, at Nyquist rate, the DAC can still maintain over 60dB SFDR, which is much higher than that of a DAC designed to achieve 14-bit static linearity with the same process without using any trimming and calibration. Without calibration, the high static linearity is achieved at the expense of conversion rate and high-frequency performance due to the large current arrays and complex switching scheme used for the matching purpose. The rapid degradation of SFDR at high-frequency band can be seen in many uncalibrated DAC designs [12][13][23][24]. Calibration reduces the current array and simplified the routing significantly, therefore reduces the parasitic effect and improve the high-frequency dynamic range while maintaining the high static performance.

Figure 4.22 compares the SFDR of this work to several prior arts, where the sampling rate is 100MHz. The DACs reported in [12] and [25] are uncalibrated and the DACs reported in [21] and [57] are self-calibrated. It is shown that as the signal frequency increases, the SFDR of this work drops much slower than those of the prior arts except the DAC reported in [21] where a track/attenuation output stage is used. This output stage attenuates the outputs during the transition period, and therefore suppresses the nonlinear effects associated with the switching transition. The drawback is that the signal power is reduced up to a half and the SFDR in the low-frequency band may be degraded since the dynamic nonlinearity in this case are much less than they are for high-frequency signals, and the extra errors introduced by the track and attenuation is even more than that they suppressed. The track/attenuation technique can be easily integrated into our prototype to improve the high-frequency SFDR by adding two matched switches in parallel with the two external resistors and/or a single switch between the two differential output nodes. Simulation suggests that with this technique, the SFDR of this work can be improved by 10 dB at the Nyquist rate. With the small feature size of this technology, large attenuation switches can be achieved with small area.

Another feature of this DAC is the low power consumption. Since the full-scale current is 10mA, with 1.5V supply, as it is expected, the analog part of the DAC costs about 15.3mW power, which does not change much with the sampling and signal frequencies. Thanks to the low supply voltage and the small feature size of the technology, the power dissipation of the digital part is very low. For 100MHz sampling rate and 41.5MHz signal frequency, it costs only 1.44mW. As shown in Figure 4.23, compared to the prior arts, this work costs extremely small area and power dissipation while maintaining state-of-art performance.

More experimental results based on the 100 chips being tested are given in Table 4.3. Before calibration, the INL of the DAC is between 6 to15LSB, while the DNL is between 3

to 6LSB. After calibration, the INL is reduced to be less than 1.2LSB and the DNL is less than 0.6LSB. They are slightly larger than the expected 0.5LSB, because the residual errors from the uncalibrated LSB arrays and the CALDAC as well as the off-chip ADC are slightly larger than we assumed in the behavioral modeling. This also slightly impacts the low-frequency-band SFDR. On the other hand, high low-frequency-band SFDR (73dB~83dB) is maintained for sampling rate up to 180MHz, while in the high-frequency band drops, the SFDR drops quickly when the sampling rate is larger than 100MHz.

It is worth mentioning that the DACs implemented in this process are not sensitive to the temperature variation. Experimental results show that the SFDR decreases by less than 3dB when the temperature changes from -40°C to 50°C. The previous mentioned data are all measured in the room temperature. It is also shown that the DACs in this work can tolerate power supplies as low as 1.25V, the decrease of SFDR is less than 3dB.

4.6 CONCLUSIONS

In today's telecommunication and signal processing systems, it is more and more important to be able to integrate the analog circuits with the large DSPs and achieve a system-on-chip solution. However, as the process feature size keeps shrinking in the favor of digital circuits, so is the supply voltage, which places a lot of challenges for high-performance analog and mixed-signal circuit design. As a result, many traditional design techniques are no longer feasible for low-voltage environments. This chapter discusses the difficulties of designing high-performance current-steering DAC in a 0.13μ digital CMOS process with supply voltage as low as 1.5V. A new foreground DAC calibration technique is presented and the experimental results show that with this technique, 14-bit resolution and accuracy can be achieved with only 0.1mm^2 active area (excluding a 16-bit low-order Σ-Δ modulator) and 16.7mW power consumption at 100MHz sampling rate. The SFDR at

100MS/s is 82.5dB in the lower signal band and 62.5dB up to the Nyquist rate. The main idea behind this technique is that with proper calibration and optimization, the dimension of the current source arrays can be dramatically reduced at the expense of a small amount of extra calibration circuitry, and meanwhile fast settling and good dynamic performance are resulted due to the parasitic reduction.

Table4.3 Performance Summary based on the 100 chips tested

Resolution	14 bits
Full-Scale Current	10mA
INL	6~15LSB (before) 0.48~1.2LSB (after)
DNL	3~6LSB (before) 0.32~0.6LSB(after)
Single-Tone SFDR @ f_{clk}=50MHz	60~67dB(before) 77~83dB(after) @ f_{out}= 0.45MHz
	52~59dB(before) 55~64dB(after) @ f_{out}=21 MHz
Single-Tone SFDR @ f_{clk}=100MHz	59~66dB(before) 75~82dB (after) @ f_{out}= 0.9MHz
	50~57dB(before) 52~62dB(after) @ f_{out}=42 MHz
Single-Tone SFDR @ f_{clk}=150MHz	56~64dB(before) 73~81dB(after) @ f_{out}=1.4 MHz
	42~48dB(before) 44~50dB(after) @ f_{out}=63 MHz
Two-Tone SFDR @ f_{clk}=100MHz	66dB(after) @ f_{out1}=23.5 MHz, f_{out2}=24.5 MHz,
Maximum Sampling Rate	180MHz
SFDR Temperature Variation	-3dB (-40°C~50°C)
Minimum Supply Voltage	1.25V
Power Dissipation @ 1.5V Supply	16.7mW @ f_{clk}= 100MHz, f_{out}=42MHz
Active Area	0.1mm^2 in 0.13μm CMOS process

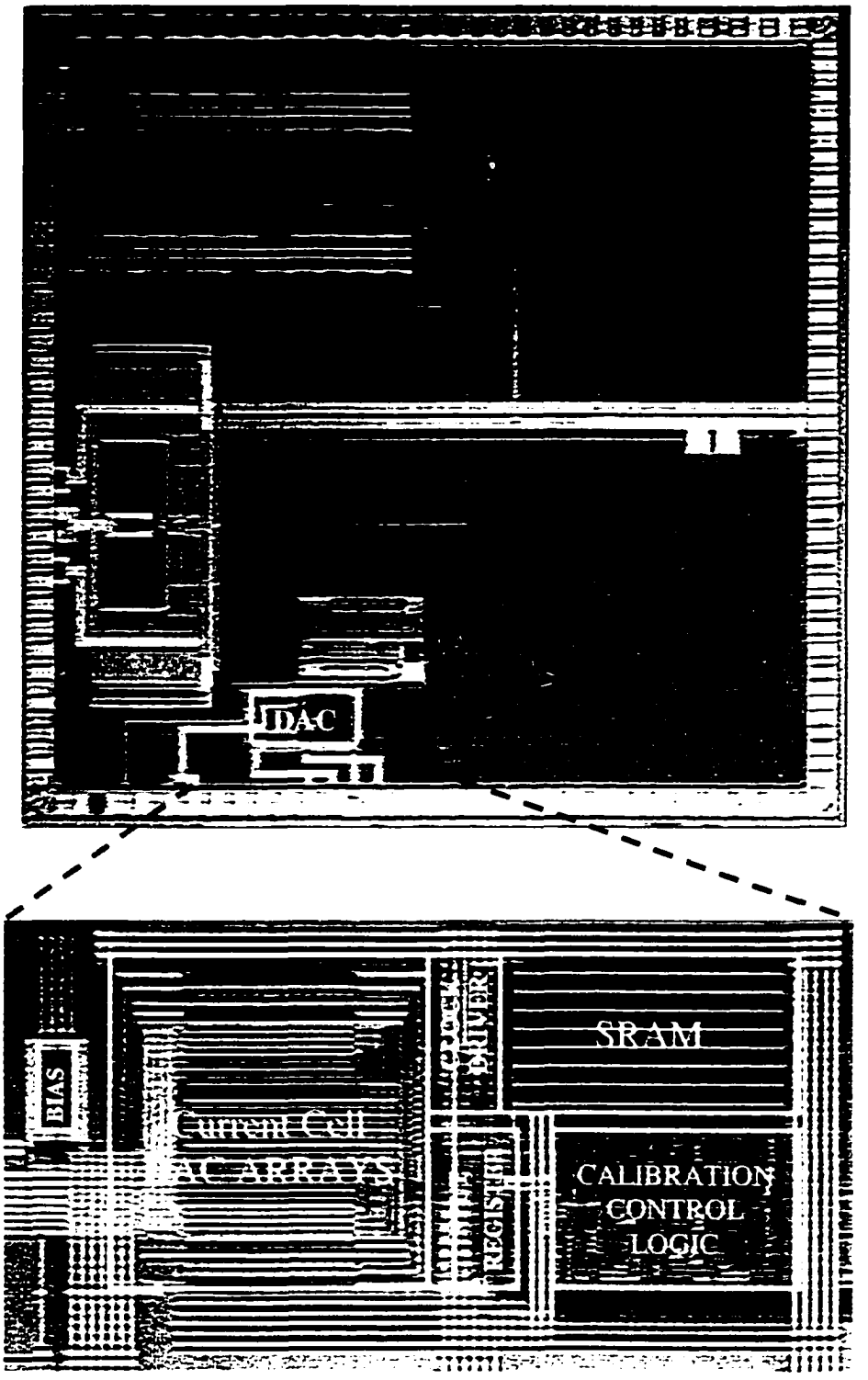


Figure 4.14 Microphotograph of the DAC

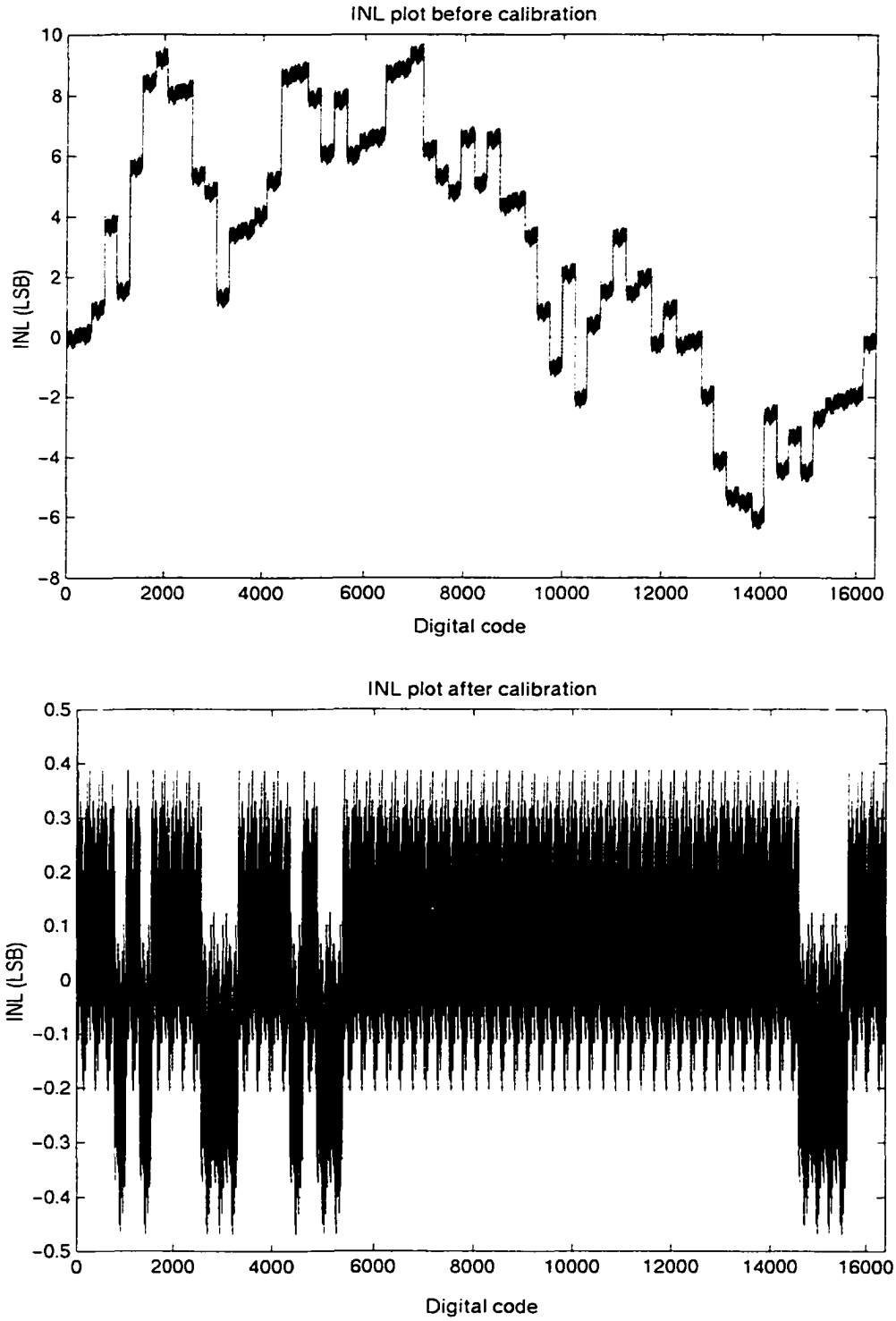


Figure 4.15 Measured INL before and after calibration

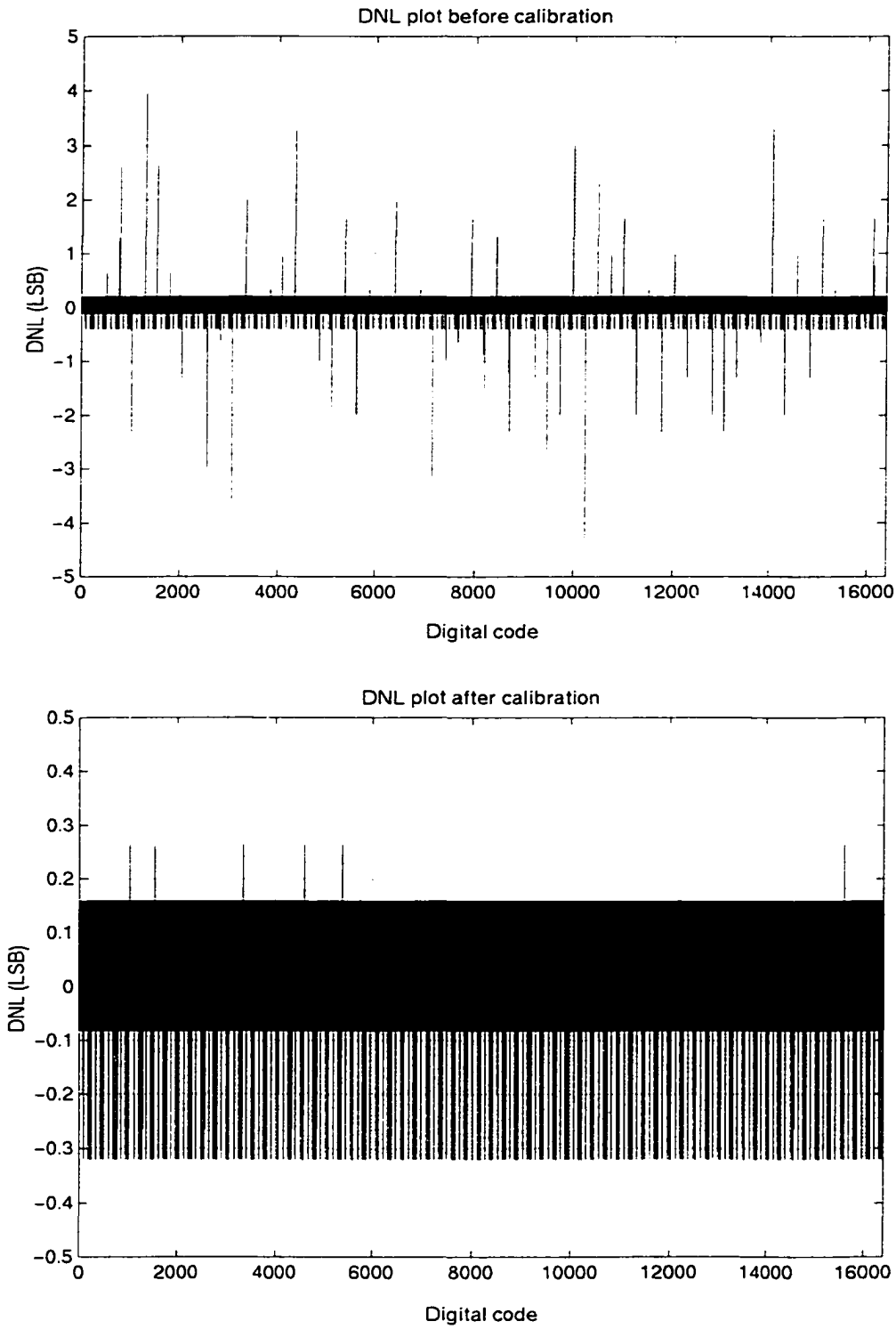


Figure 4.17 Measured DNL before and after calibration

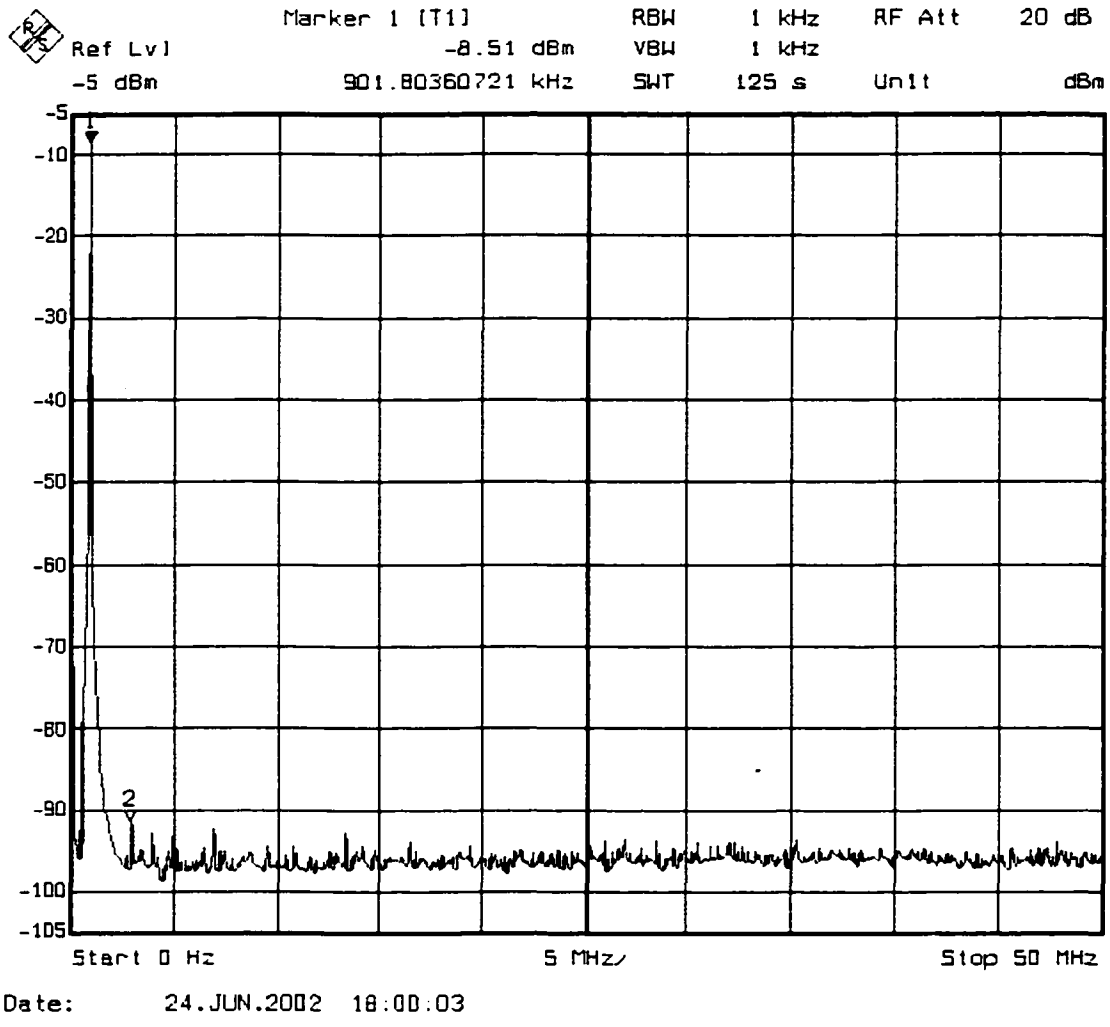


Figure 4.17 Output spectrum of a single tone at the lower signal band (f_{clk}=100MHz, f_{out}= 0.9MHz)

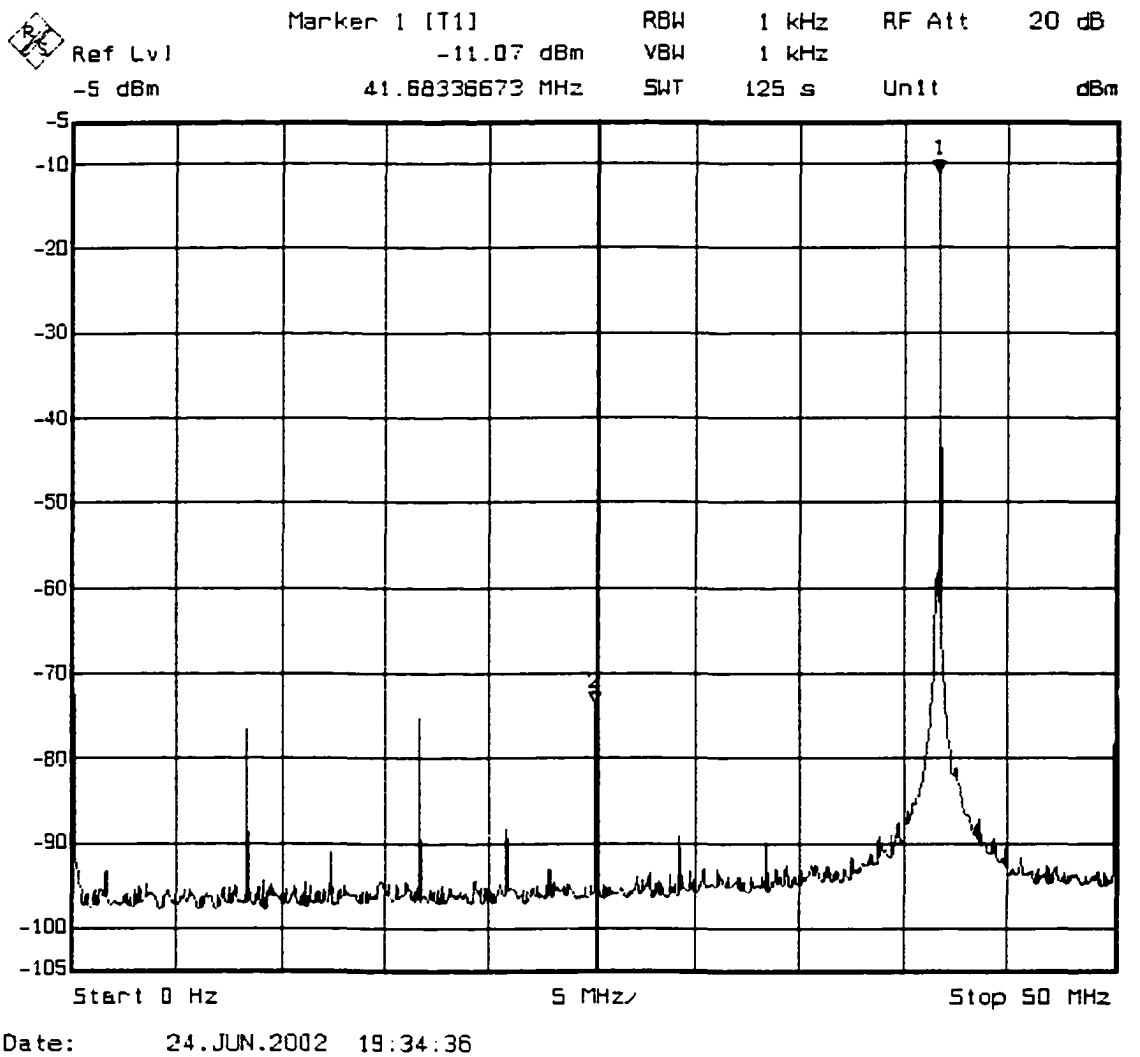


Figure 4.18 Output spectrum of a single tone at the higher signal band (f_{clk}=100MHz, f_{out}= 41.7MHz)

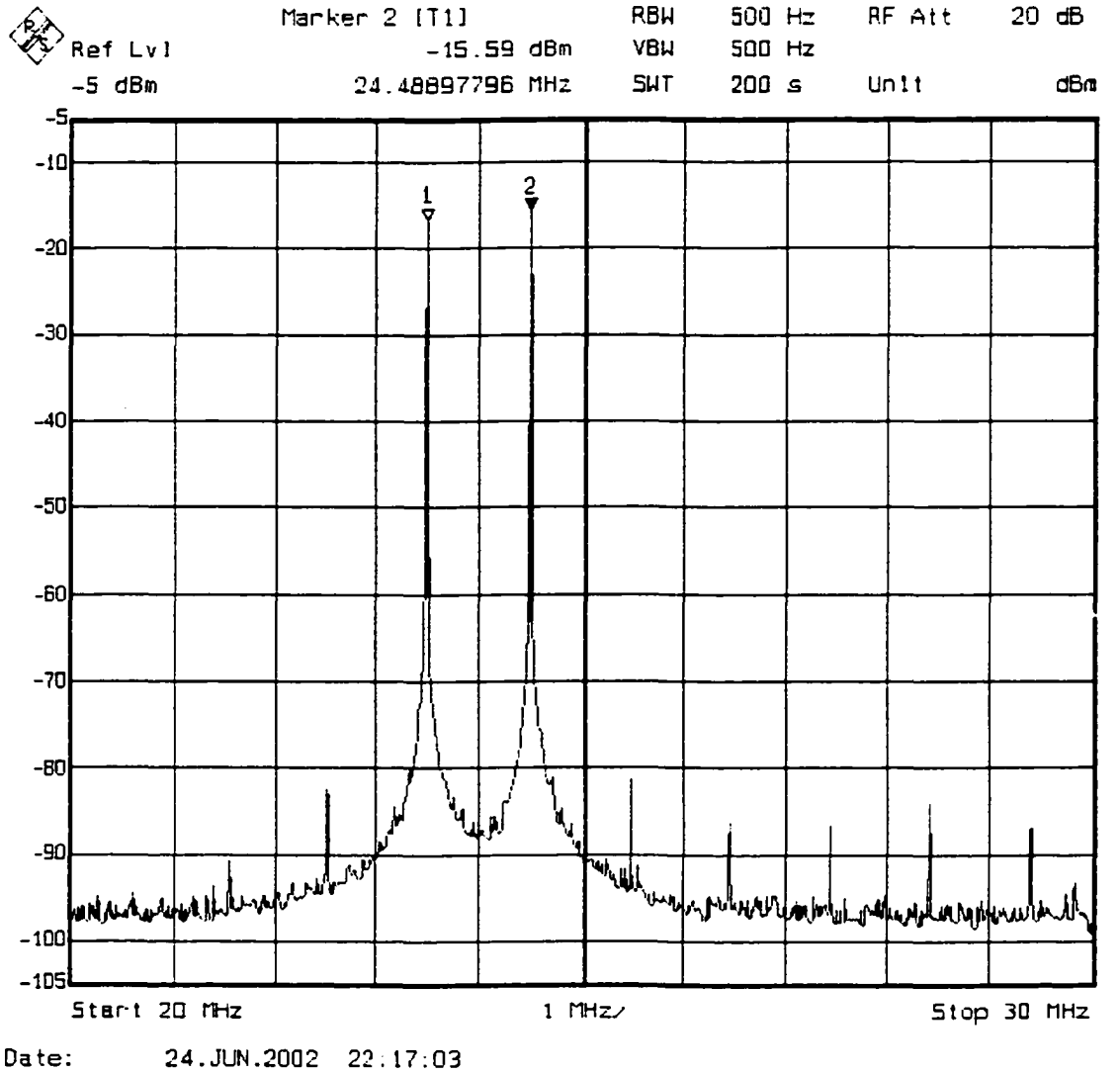


Figure 4.19 Output spectrum of two tones
(f_{clk}=100MHz, f_{out1}= 23.5MHz, f_{out2}= 24.5MHz)

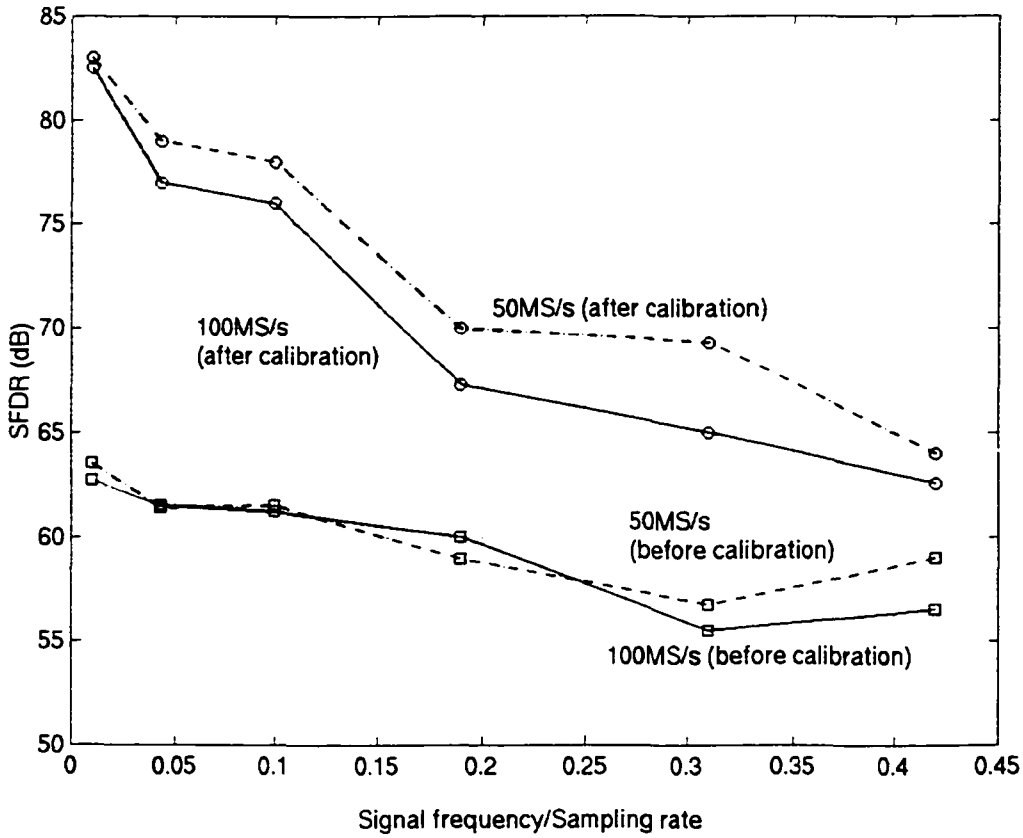


Figure 4.20 Summary of the full-scale output SFDR

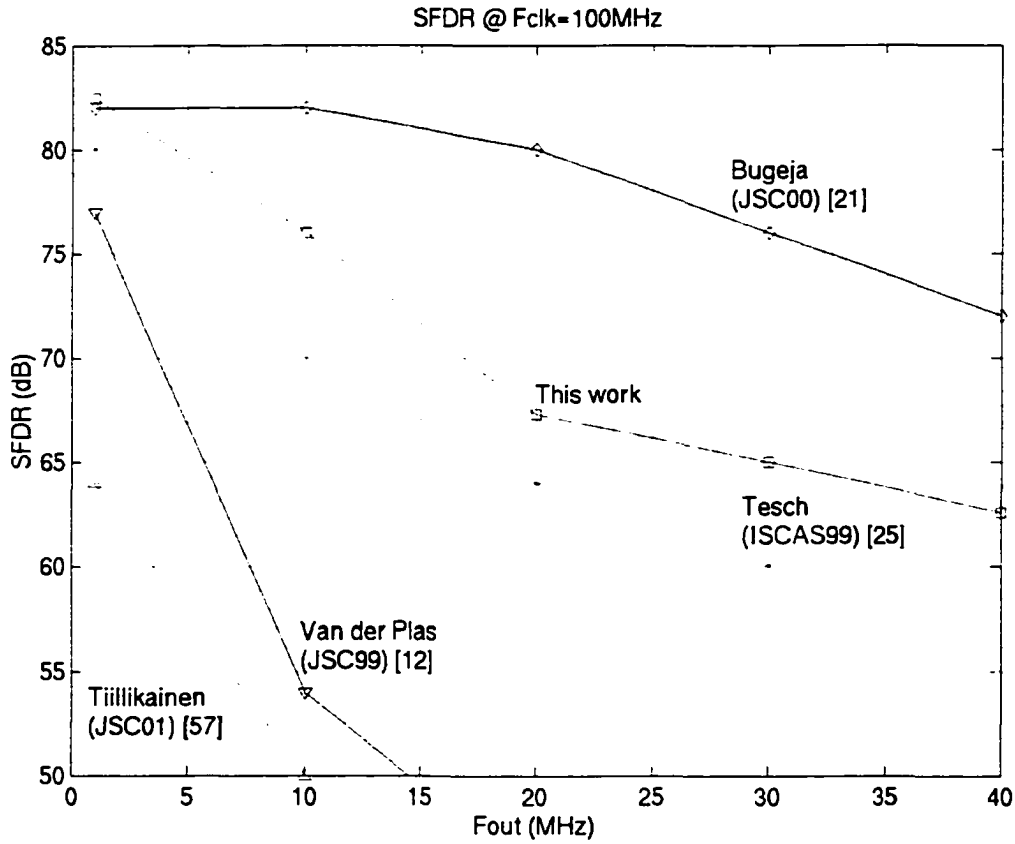


Figure 4.22 Comparison of SFDR to the prior arts (f_{clk}=100MHz)

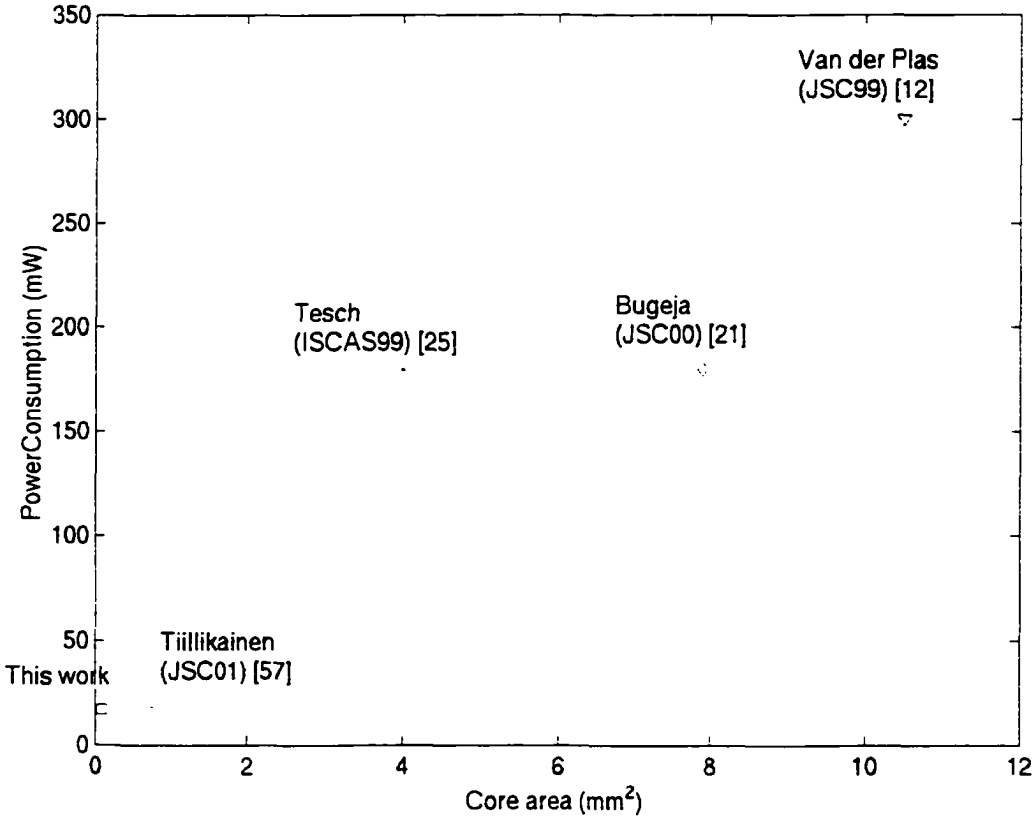


Figure 4.23 Comparison of area and power dissipation to the prior arts

REFERENCES

[1] J. Vankka, J. Ketola, O. Vaananen, J. Sommarek, M. Kosunen, and K. Halonen, "A GSM/EDGE/WCDMA modulator with on-chip D/A converter for base station", *IEEE Int. Solid-State Circuits Conf., Digest of Technical Papers*, pp.236-237, 2002.

[2] I. Mehr, M. Prabir, and D. Paterson, "A 12-bit Integrated Analog Front End for Broadband Wireline Networks," *IEEE J. Solid-State Circuits*, vol.37, pp.302-309, May, 2001.

[3] W. Claes, W. Sansen, and A. Puers, "A 40-μA/Channel Compensated 18-Channel Strain Gauge Measurement System for Stress Monitoring in Dental Implants," *IEEE J. Solid-State Circuits*, vol.37, pp.293-301, March 2002.

[4] C. Yang, V. Stojanovic, S. Modjtahedi, M. Horowitz, and W. Ellersick, " A Serial-Link Transceiver Based on 8-G Samples/s A/D and D/A Converters in 0.25-μm CMOS," *IEEE J. Solid-State Circuits*, vol.36, pp.293-301, Nov. 2001.

[5] C.-H. Lin and K. Bult, "A 10b, 500MSample/s DAC in 0.6 mm^2", *IEEE J. Solid-State Circuits*, vol.33, pp.1948-1958, Dec. 1998.

[6] J. Wikner and N. Tan, "Modeling of CMOS Digital-to-Analog Converters for Telecommunication," *IEEE Trans. Circuits and Systems II*, vol.46, pp.489-499, May, 1999.

[7] D. Johns and K. Martin, *Analog Integrated Circuit Design*, Wiley, New York, 1997.

[8] B. Razavi, *Principles of Data Conversion System Design*, IEEE Press, Piscataway, NJ, 1995.

[9] M. Pelgrom, A. Duinmaijer, and A. Welbers,. "Matching Properties of MOS Transistors", *IEEE J. Solid-State Circuits*, vol.SC-24, pp.1433-1439, Oct. 1989.

[10] A. Van den Bosch, M. Steyaert, and W. Sansen, "The Extraction of Transistor Mismatch Parameters: The CMOS Current-Steering D/A Converter as A Test Structure", *IEEE Int. Conf. Symp. Circuits and Systems*, vol.II, pp.745-748, 2000.

[11] J. Bastos, M. Steyaert, and W. Sansen, "A High Yield 12-bit 250-MS/s CMOS D/A Converter," *Proc. IEEE 1996 Custom Integrated Circuits Conf.*, pp.431-434, May 1996.

[12] G. Van der Plas, J. Vandenbussche, W. Sansen, M. Steyaert, and G. Gielen, "A 14-bit intrinsic accuracy Q^2 Random Walk CMOS DAC," *IEEE J. Solid-State Circuits*, vol.34, pp.1708-11718, Dec. 1999.

[13] J. Bastos, A. Marques, M. Steyaert, and W. Sansen, "A 12 bit Intrinsic accuracy High Speed CMOS DAC", *IEEE J. Solid-State Circuits*, vol.33, pp.1959-1969, Dec. 1998.

[14] Y. Nakamura, T. Miki, A. Maeda, H. Kondoh, and N. Yazawa, "A 10-bit 70MS/s CMOS D/A Converter," *IEEE, J. Solid-state Circuits*, vol.26, pp.637-642, Apr. 1991.

[15] H. Schouwenaas, D. Groeneveld, and H. Termeer, "A Low-Power Stereo 16-bit CMOS D/A Converter for Digital Audio", *IEEE J. Solid-state Circuits*, vol.23, pp.1290-1297, Dec. 1988.

[16] J. Bastos, M. Steyaert, A. Pergoot, and W. Sansen, "Influence of die attachment on MOS transistor matching," *IEEE Trans.Semiconduct. Manufact.*, pp.209-217, May 1997.

[17] T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, and Y. Horiba, "An 80-MHz 8-bit CMOS D/A converter," *IEEE, J. Solid-state Circuits*, vol.SC-21, pp.983-988, Dec. 1986.

[18] A. Van den Bosch, M. Steyaert, and W. Sansen,"SFDR-bandwidth limitations for high-speed high-resolution current-steering CMOS D/A converters," *Proc. IEEE Int. Conf. Electronics, Circuits and Systems*, pp.1193-1196, 1999.

[19] A. Van den Bosch, M. Borremans, M. Steyaert, and W. Sansen, "A 10-bit 1-GSample/s Nyquist current-steering CMOS D/A converter," *IEEE J. Solid-State Circuits*, vol.36, pp.315 –324, March 2001.

[20] A.Rugeja and B.Song, "A 14-b, 100-MS/s CMOS DAC Designed for Spectral Performance," *IEEE J. Solid-state Circuits*, vol.34, pp.1719-1731, Dec. 1999.

[21] A.Rugeja and B.Song, "A Self-Trimming 14-b 100-MS/s CMOS DAC," *IEEE J. Solid-state Circuits*, vol.35, pp.1841-1852, Dec. 2000.

[22] J. Wikner and N. Tan, "Influence of Circuit Imperfections on the Performance of DACs," *Analog Integrated Circuits and Signal Processing*, pp.7-20, Jan. 1999.

[23] A. Van den Bosch, M. Borremans, J. Vandenbussche, G. Van der Plas, A. Marques, J. Bastos, M. Steyaert, G. Gielen, and W. Sansen, "A 12-bit 200MHz Low Glitch CMOS D/A converter," *IEEE Conf. Custom Integrated Circuits*, pp.249 –252, 1998.

[24] A. Van den Bosch, M. Borremans, M. Steyaert, and W. Sansen, "A 12b 500Msample/s Current-Steering CMOS D/A converter," *IEEE Int. Solid-State Circuits Conf., Digest of Technical Papers*, pp.366-367, 2001.

[25] B. Tesch, P. Pratt, K. Bacrania, and M. Sanchez, "A 14-b, 125MSPS Digital-to-Analog Converter and Bandgap Voltage Reference in 0.5μm CMOS," *IEEE Intl. Symp. Circuits and Systems*, vol.II, pp.452-455, 1999.

[26] B. Tesch and J. Garcia, "A Low Glitch 14-b 100-MHz D/A Converter," *IEEE Int. Solid-State Circuits Conf., IEEE J. Solid-state Circuits*, vol.32, pp.1465-1469, Sep. 1997.

[27] J. Park, S. Lee, J. Yoon, and S. Lee, " A 3V 10b 70MHz Digital-to-Analog Converter for Video Applications," *Proc. the First IEEE Asia Pacific Conf. on ASICs*, pp.186-189, 1999.

[28] J. Park, S. Lee, J. Yoon, and S. Lee, " A 3V 10b 100MHz Digital-to-Analog Converter for Cable Modem Applications," *IEEE Trans. Consumer Electronics*, vol.46, pp.1043-1047, Nov. 2000.

[29] N. Kumazawa, N. Fukushima, N. Ono, and N. Sakamot, "An 8bit 150MHz CMOS D/A Converter with 2Vp-p Wide Range Output," *IEEE Symp. VLSI Circuits*, pp.55-56, 1990.

[30] H. Takakura, M. Yokoyama, and A. Yamaguchi, "A 10bit 80MHz Glitchless CMOS D/A Converter," *IEEE Conf. Custom Integrated Circuits*, pp.26.5.1-26.5.4, 1991.

[31] C. Bastiaansen, D. Groeneveld, and H. Sshouwenaars, "A 10-b 40-MHz 0.8-μm CMOS Current-Output D/A Converter," *IEEE J. Solid-State Circuits*, vol.26, pp.917 –921, July 1991.

[32] D. Mercer, "A 16-b D/A Converter with Increased Spurious Free Dynamic Range," *IEEE J. Solid-state Circuits*, vol.29, pp.1180-1185, Oct. 1994.

[33] H. Kohno, Y. Nakamura, A. Kondo, H. Amishiro, T. Miki, and K. Okada, "A 350-MS/s 3.3-V 8-bit CMOS D/A Converter Using a Delayed Driving Scheme," *IEEE Conf. Custom Integrated Circuits*, pp.10.5.1-10.5.4, 1995.

[34] D. Mercer and L. Singer, "12-b 125MSPS CMOS D/A Designed for Spectral Performance," *Proc. Int. Symp. Low Power Electronics and Design*, pp.243-246, 1996.

[35] A. Cremonesi, F. Maloberti and G. Polito, "A 100-MHz CMOS DAC for Video-Graphic Systems," *IEEE J. Solid-state Circuits*, vol.24, pp.635-639, June, 1989.

[36] S. Chin and C. Wu, "A 10-b 125-MHz CMOS Digital-to-Analog Converter (DAC) with Threshold-Voltage Compensated Current Sources," *IEEE J. Solid-state Circuits*, vol.29, pp.1374-1380, Nov. 1994.

[37] T. Kamoto, Y. Akazawa, and M. Shinagawa, "An 8-bit 2-ns Monolithic DAC," *IEEE J. Solid-state Circuits*, vol.23, pp.142-146, Feb. 1988.

[38] N. Tan, "A 1.5-V 3-mW 10-bit 50-Ms/s CMOS DAC with Low Distortion and Low Intermodulation in Standard Digital CMOS Process", *IEEE Conf. Custom Integrated Circuits*, pp.27.5.1-27.5.4, 1997.

[39] Y. Cong and R.Geiger, "Formulation of INL and DNL Yield Estimation in Current-Steering D/A Converters," *IEEE Intl. Symp. Circuits and Systems*, vol.III, pp.149-152, 2002.

[40] K. Lakshimikumar and al., "Characterization and Modeling of Mismatch in MOS Transistors for Precision Analog Design," *IEEE J. of Solid-State Circuits*, Dec. 1986,vol.21, pp.1057 –1066.

[41] K. Lakshimikumar and al., "Reply to 'A Comment on 'Characterization and Modeling of Mismatch in MOS Transistors for Precision Analog Design'," *IEEE J. Solid-State Circuits*, Feb. 1988,vol.23, pp.296.

[42] C. Conroy and al. "A Comment on 'Characterization and Modeling of Mismatch in MOS Transistors for Precision Analog Design'," *IEEE J. Solid-State Circuits*, vol.23, pp.294-296, Feb. 1988.

[43] A. Van den Bosch and al., "An Accurate Statistical Yield Model for CMOS Current-Steering D/A Converters," *IEEE Intl. Symp. Circuits and Systems*, vol.4, pp.105-108, 2000.

[44] M. Borremans, A. Van den Bosch, M. Steyaert, and W. Sansen,"A Low Power, 10-bit CMOS D/A Converter for High Speed Applications," *IEEE Conf. Custom Integrated Circuits*, pp.157 –160, 2001.

[45] Y. Cong and R.Geiger, "Switching Sequence Optimization for Gradient error Compensation in Thermometer-Decoded DAC Arrays", *IEEE Trans. Circuits and Systems II*, vol.47, pp.585-595, July 2000.

[46] M. Song, B. Kang and E. Joe, "A 10-bit 80MHz 3.0V CMOS D/A Converter for Video Applications," *IEEE Trans. Consumer Electronics*, vol.46, pp.1043-1047, Nov. 2000.

[47] B. Henriques and J. Franca, "A High-Speed Programmable CMOS Interface System Combining D/A Conversion and FIR Filtering," *IEEE J. Solid-state Circuits*, vol.29, pp.972-977, Aug. 1994.

[48] B. Henriques, K. Kananen, J. Franca, and J. Rapeli, "A 10 Bit Low-Power CMOS D/A Converter with On-Chip Gain Error Compensation," *IEEE Conf. Custom Integrated Circuits*, pp.10.6.1 –10.6.4, 1995.

[49] T. Wu, C. Jih, J. Chen, and C. Wu, "A Low Glitch 10-bit 75-MHz CMOS Video D/A Converter," *IEEE J. Solid-state Circuits*, vol.30, pp.68-72, Jan. 1995.

[50] L. Sumanen, M. Waltari, and K. Halonen, "A 10-bit High-Speed Low-Power CMOS D/A Converter in 0.2mm^2," *Proc. IEEE Int. Conf. Electronics, Circuits and Systems*, vol.1, pp.15-18, 1998.

[51] D. Groeneveld, H. Schouwenaars, H. Termeer, and C. Bastiaansen, "A Self-Calibration Technique for Monolithic High-Resolution D/A Converters," *IEEE J. Solid-state Circuits*, vol.24, pp.1517-1522, Dec. 1989.

[52] F. DelPova, et al. "Digital-Analog Converter with Self-Calibration Current Sources," US Patent 5870044, Feb., 1999.

[53] D. Groeneveld, et al. "Calibrated Current Source with Ripple Reduction," US Patent 5021784, June 4, 1991.

[54] S. Daubert, et al. "Self-Calibrating High Speed D/A Converter", US Patent 5646619, July 8, 1997.

[55] R. van de Plassche, "Current Source Arrangement Having a Precision Current-Mirror Circuit," US Patent 4573005, Feb. 25,1986.

[56] M. Storey, "Digital-to-Analogue Converter Including Calibrated Current Sources," US Patent 4647907, March 3, 1987.

[57] M. Tiilikainen, "A 14-bit 1.8-V 20-mW 1-mm^2 CMOS DAC," *IEEE J. Solid-state Circuits*, vol.36, pp.1144-1147, July 2001.

[58] A. Tang and C. Toumazou, "Novel Self-Calibrated High-Speed D/A Converter Using Trimmable Current Sources," *IEEE Intl. Symp. Circuits and Systems*, vol.5, pp.469-472, 1994.

[59] G. Miller, M. Coln, L. Singer and P. Oaklander, "A True 16b Self-Calibrating BiCMOS DAC," *Proc. IEEE Int. Solid-State Circuits Conf.*, pp58-59, 1993.

[60] Y. Pang, et al. "Error Correction System for Digital to Analog Converters," US Patent 4967197, Oct. 30, 1990.

[61] J. Fairchild, et al. "16 Bit Error-Correcting Digital-to-Analog Converter," US Patent 5153592, Oct. 6, 1992.

[62] J. Gersbach, "Self Calibration Segmented Digital-to-Analog Converter," US Patent 5666118, Sep. 9, 1997.

[63] A. Sedra and K. Smith, "Microelectronic Circuits," Oxford University Press, New York, 1997.

[64] "3V/5V, 450μA 16-bit Sigma Delta ADC", Datasheet, Analog Devices, Inc, Wilmington, MA, 2000.

[65] V. Peluso, M. Steyaert, and W. Sansen, "A 1.5-V-100-μW ΔΣ Modulator with 12-b Dynamic Range Using the Switched-Opamp Technique," *IEEE J. Solid-state Circuits*, vol.32, pp.943-952, July, 1997.

[66] A. Coban and P. Allen, "A 1.5V 1.0mW Audio ΔΣ Modulator with 98dB Dynamic Range," *IEEE Int. Solid-State Circuits Conf., Digest of Technical Papers*, pp.50-51, 1999.

[67] M. Dessouky, A. Kaiser, M. Louerat and A. Greiner, "Analog Design for Reuse-Case Study: Very Low-voltage ΔΣ Modulator," *Proc. Conference and Exhibition Design, Automation and Test in Europe*, pp.353-360, 2001.

[68] M. Dessouky and A. Kaiser, "Very Low-Voltage Digital-Audio ΔΣ Modulator with 88-dB Dynamic Range Using Local Switch Bootstrapping," *IEEE J. Solid-state Circuits*, vol.36, pp.349-355, March, 2001.

[69] T. Tille, J. Sauerbrey and D. Schmitt-Landsiedel, "A 1.8-V MOSFET-Only ΔΣ Modulator Using Substrate Biased Depletion-Mode MOS Capacitors in Series Compensation," *IEEE J. Solid-state Circuits*, vol.36, pp.1041-1047, July, 2001.

[70] S. Rabii and B. Wooley, "A 1.8-V Digital-Audio Sigma-Delta Modulator in 0.8-μm CMOS," *IEEE J. Solid-state Circuits*, vol.32, pp.783-796, June, 1997.

[71] O. Bajdechi and J. Huijsing, "A 1.8-V ΔΣ Modulator Interface for an Electet Microphone with On-Chip Reference," *IEEE J. Solid-state Circuits*, vol.37, pp.279-285, March, 2002.

ACKNOWLEDGMENTS

The past four and a half years of Ph.D. study has been a very special experience that I will often cherish in my life. I believe any achievements I have made can not be accomplished without the help of many people.

I would like to express my greatest gratitude to my advisor, Professor Randall Geiger. The combination of his breath and depth of knowledge in circuits and systems provided me with a unique and rewarding experience. I would like to thank him for the many inspiring conversation and advice on research and teaching, and generous help to my life and career. I especially appreciate him for always giving me confidence and courage.

I aiso owe a lot of appreciation to Mr. Doug Garrity for his strong support on my project: a 1.5V 100MS/s self-calibrated DAC. I thank him for giving me such a precious opportunity to do a one and half year of internship in the semiconductor product sector of Motorola Co., and sponsoring me in VerilogA training and conferences attendance.

I would like to thank my colleagues from ISU, Huiting Chen, Mao-Feng Lan, Saqib Marlik, Kumar Parthasarathy, Mark Schlarmann, Yonghui Tang, Jie Yan and colleagures from Motorola, Brandt Braswell, Bruce Newman, Dana Price, David Locascio, Merit Hong, Mike McGowan, Rick Sherman, Patrick Shen, Pham Thuy, Rick Riley and Shirley Hall for their helpful discussions. I especially appreciate Saqib marlik and Yonghui Tang for their time spent on software maintenance, Bruce Newman for helping me with the layout synthesis of the calibration circuitry and setting up Cadence working environment, and Dana Price and Rick Sherman for their patience and support during the testing.

I would like to thank Professor Degang Chen, Professor Chris Chu, Professor Robert Weber and Professor Yuhong Yang for being on my program of study committee.

I also would like to thank for the support and encouragement from my parents, Xianming Gong and Jiangzi Cong, and my sister Lin Cong. Their love have been the source of my strength.

Last but not least, I would like to thank my husband Li Liu. No word can express my gratitude for his unconditional support and love through the most stressful and exhausting times of my study. His understanding and encouragement is the most comfort and strongest motivation to me. His knowledge in DSP theory and Lab instruments are also very helpful to my research. For his great sacrifice and extreme patience in enduring our five years of separation, I would like to dedicate this dissertation to my husband Li Liu.